intel

ICE™-51/ICE™-44 IN-CIRCUIT EMULATOR COMMAND DICTIONARY

Intel Corporation, 3065 Bowers Avenue,
Santa Clara, California 95051 Order Number: 9801005-02

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

AEDIT	iLBX	iOSP	MULTIBUS
BITBUS	im	iPDS	MULTICHANNEL
BXP	iMMX	iRMX	MULTIMODULE
COMMputer	Insite	iSBC	Plug-A-Bubble
CREDIT	Intel	iSBX	PROMPT
i	IntelBOS	iSDM	Promware
iATC	Intelevision	iSXM	Ripplemode
ICE	intelligent Identifier	Library Manager	RMX/80
iCS	intelligent Programming	MCS	RUPI
iDBP	Intellec	Megachassis	System 2000
iDIS	Intellink	MICROMAINFRAME	UPI
I^2ICE			

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

*MULTIBUS is a patented Intel bus.

PREFACE

This manual lists all the user commands for the ICE™-51/-ICE™-44 In-Circuit Emulator in alphabetical order by the first entry in the command. The Dictionary also lists some terms (e.g., *Expression)* that are part of the definitions of several different commands, in order to avoid lengthy duplication. The Dictionary assumes the reader is familiar with the explanations of the commands given in the following manual:

> *The ICE-51/44 In-Circuit Emulator Operating Instructions for ISIS User's,* Manual Order Number 9801004.

The reader may also wish to consult the following related publications:

> *Getting Started With the ICE™-51 In-Circuit Emulator,* Manual Order Number 121595.
>
> *Intel MCS-51™ Family User's Manual,* Manual Order Number 121517.

Command Format Notation

Command format notation is a kind of diagram for ICE commands. The notation shows what command words to use, indicates parts of the command that can be omitted or included at your option, and shows the places in the command where you have a choice among several kinds of entry.

Keywords and User Entries

A keyword is a command word with a fixed spelling and interpretation defined by the system. In the notation keywords are shown as ALL CAPS. All keywords may be abbreviated to the first three characters, and several may be shortened to one or two characters. The abbreviated forms of keywords are not shown in the format notation, but are used in some of the examples.

A user entry is a word or hyphenated phrase shown in *lowercase italic*. A user-entry represents a class of possible entries that contain too many variations to allow listing them all explicitly. The lower-case items themselves are not part of the command language.

Here is an example of the notation for keywords and user entries:

EVALUATE *expression*

In this command format, EVALUATE is a keyword; it cannot be omitted from the command or spelled differently although it may be abbreviated. The term *expression* represents all the possible ways of obtaining a numeric result including single entries and longer formulas.

Required and Optional Entries

Required entries are given in the notation without any enclosure. In the format EVALUATE *expression*, both the keyword EVALUATE and an entry corresponding to *expression* are required to make a valid command.

Optional entries are enclosed in square brackets. Any entry shown in square brackets can be omitted leaving a valid command. Here is an example of a command format containing required and optional entries:

LOAD :F*n*:*filename* [NOCODE] [NOSYMBOLS]

Repeatable Entries

Entries that can be repeated at user option are enclosed in square brackets and followed by an ellipsis (three adjacent periods). The most common use of this notational form is to represent lists of entries separated by commas. Here is an example of a format containing a list:

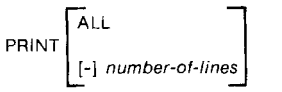
REMOVE MACROS [*macro-name* [, *macro-name*] ...]

In this command format, the list of macro names can be omitted entirely, or it can have just one macro name, or it can be several macro names separated by commas.

Choices of Entries

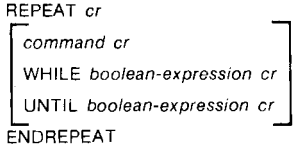
A choice of entries is indicated by a vertical list of the entries enclosed in curved braces or square brackets; such a list is called a "menu". A menu enclosed in square brackets means "select none or one". For example:

Format



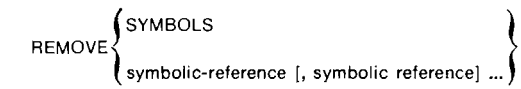
The menu shows that you can omit the entry after PRINT, or you can select the keyword ALL, or you can enter a "number of lines" optionally with a minus sign preceding it.

A menu enclosed in square brackets and followed by an ellipsis (...) means "select none, one, or more than one, in any order". Here is an example of this kind of format:

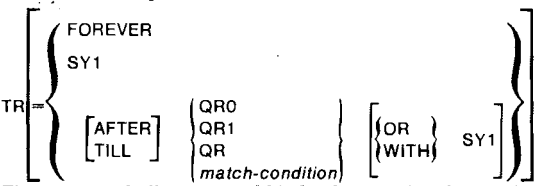


A menu enclosed in curved braces means "select one and only one". In the following example, the choice is between the keyword SYMBOLS and a list of symbolic references. One of these two kinds of entries must be included to form a valid command.

Format



A menu can contain entries that themselves contain menus. Here is an example of such a command format:



This command allows several kinds of entry after the equals sign:

1. The keyword FOREVER.
2. The keyword SY1.
3. One of the keywords QR0, QR1, or QR.
4. An entry specifying a matching condition.
5. The entries described in 3 and 4 can be preceded by one of the keywords AFTER or TILL.

Single-Character Entries

The emulator command language contains single-character entries that serve as delimiters, punctuation marks, operators, and other uses. These are shown in **bold** in the command formats.

For example:

:macro-name

Other Notation Signs

cr Intermediate carriage return; ends input line but because of its place in the command it does not terminate the command.

; A semicolon precedes a comment in a command format or example.

This section gives all the emulator keywords, their abbreviations, and the command entries in this dictionary where they appear.

Keyword	Abbreviation	Dictionary Entries
ADDR	ADD	Match Condition; Trace Reference; Unlimited Match Condition.
AFTER	AFT	TR Commands.
ALL	ALL	PRINT Command.
AND	AND	Expression; Match Condition; Unlimited Match Condition.
ASCII	ASC	BASE Command.
ASM	A	ASM Command.
BASE	BAS	BASE Command.
BR	BR	BR, BR0, BR1 Commands; GO Command; GR Command; RESET Command.
BR0	BR0	BR, BR0, BR1 Commands; GO Command; GR Command; RESET Command.
BR1	BR1	BR, BR0, BR1 Commands; GO Command; GR Command; RESET Command.
BUFFERSIZE	BUF	BUFFERSIZE Command; Expression.
CAUSE	CAU	CAUSE Command; Expression.
CBYTE	CBY	CBYTE Commands; Expression.
CHIP	CHI	RESET Commands.
COUNT	C	COUNT Command.
CYC	CYC	Trace Reference.
DASM	D	DASM Command.
DATA	DAT	Trace Reference.
DBYTE	DBY	DBYTE Commands; Expression.
DEFINE	DEF	DEFINE Macro Command; DEFINE Symbol Command.
DIR	DIR	DIR Command.
DISABLE	DIS	DISABLE Commands.
DMUX	DMU	Trace Reference.
DPTR	DPT	DPTR Command; Expression.
ELSE	ELS	IF Command.
EM	EM	DEFINE Macro Command.
ENABLE	ENA	ENABLE Commands.
ENDCOUNT	END	COUNT Command.
ENDIF	END	IF Command.
ENDREPEAT	END	REPEAT Command.
ERROR	ERR	ERROR Command.

Keyword	Abbreviation	Dictionary Entries
EVALUATE	E	EVALUATE Command.
EXIT	EXI	EXIT Command.
EXPANSION	EXP	DISABLE Commands; ENABLE Commands.
FOREVER	FOR	GO Command; GR Command; TR Command.
FRAME	FRA	Trace Mode Command; Trace Reference.
FROM	F	GO Command; STEP Command.
GO	G	GO Command.
GR	GR	GR Command.
H	H	BASE Command; SUFFIX Command.
HELP	H	HELP Command.
HTIMER	HTI	Expression; HTIMER Command.
ICE	I	RESET Commands.
IF	IF	IF Command.
INCLUDE	INC	INCLUDE Command.
INSTRUCTION	INS	TRACE Mode Command.
INTERRUPT	INT	INTERRUPT Command.
IS	IS	Match Condition; Unlimited Match Condition.
LATCH	LAT	DISABLE Commands; ENABLE Commands.
LENGTH	L	Partition.
LIST	LIS	LIST Command.
LOAD	LOA	LOAD Command.
LOCATION	LOC	Match Condition; Unlimited Match Condition.
MACRO	MAC	MACRO Display Command; PUT Command. REMOVE Command.
MAP	MAP	MAP Command.
MOD	MOD	Expression.
MOVE	M	MOVE Command.
NEWEST	N	NEWEST Command.
NOCODE	NOC	LOAD Command; SAVE Command.
NOSYMBOLS	NOS	LOAD Command; SAVE Command.
NOT	NOT	Expression.
O	O	BASE Command; SUFFIX Command.
OLDEST	O	OLDEST Command.
OPCODE	OPC	Expression; Match Condition; OPCODE Command; Unlimited Match Condition.
OR	OR	Expression; GO Command; GR Command; TR Command.

Keyword	Abbreviation	Dictionary Entries
ORG	ORG	ASM Command.
ORIF	ORI	IF Command.
OUT	OUT	DISABLE Command; ENABLE Command.
PBYTE	PBY	Expression; PBYTE Commands.
PC	PC	Expression; PC Command.
PPC	PPC	Expression; PPC Command.
PRINT	P	PRINT Command.
PUT	PUT	PUT Command.
P0	P0	Match Condition; Trace Reference; Unlimited Match Condition.
P1	P1	Match Condition; Trace Reference; Unlimited Match Condition.
P2	P2	Match Condition; Trace Reference; Unlimited Match Condition.
Q	Q	BASE Command; SUFFIX Command.
QR	QR	QR, QR0, QR1 Commands; RESET Command; TR Command.
QR0	QR0	QR, QR0, QR1 Commands; RESET Command; TR Command.
QR1	QR1	QR, QR0, QR1 Commands; RESET Command; TR Command.
RBIT	RBI	Expression; RBIT Commands.
RBS	RBS	Expression; RBS Command.
RBYTE	RBY	Expression; RBYTE Commands.
REGISTERS	R	REGISTERS Command.
REMOVE	REM	REMOVE Command.
REPEAT	REP	REPEAT Command.
RESET	RES	RESET Commands.
R0	R0	Expression; R0 - R7 Commands.
R1	R1	Expression; R0 - R7 Commands.
R2	R2	Expression; R0 - R7 Commands.
R3	R3	Expression; R0 - R7 Commands.
R4	R4	Expression; R0 - R7 Commands.
R5	R5	Expression; R0 - R7 Commands.
R6	R6	Expression; R0 - R7 Commands.
R7	R7	Expression; R0 - R7 Commands.
SAVE	SAV	SAVE Command.
SECONDS	SEC	SECONDS Command.
STACK	STA	STACK Command.
STEP	S	STEP Command.
SUFFIX	SUF	SUFFIX Command.

Keyword	Abbreviation	Dictionary Entries
SY	SY	SY, SY0, SY1 Display Commands; RESET Command.
SY0	SY0	DISABLE Commands; ENABLE Commands; Expression; SY, SY0, SY1 Display Commands; GO Command; GR Command.
SY1	SY1	DISABLE Commands; ENABLE Commands; Expression; SY, SY0, SY1 Display Commands; TR Command.
SYMBOLS	SYM	REMOVE Command; SYMBOLS Command.
SYMBOLIC	SYM	DISABLE Commands; ENABLE Commands.
T	T	BASE Command; SUFFIX Command.
THEN	THE	IF Command.
TILL	T	GO Command; GR Command; TR Command.
TIMER	TIM	Expression; TIMER Command.
TM0	TM0	Expression; TM0, TM1 Commands.
TM1	TM1	Expression; TM0, TM1 Commands.
TO	TO	Partition.
TOVF	TOV	Trace Reference.
TR	TR	TR Commands.
TRACE	TRA	TRACE Mode Commands.
UNTIL	U	COUNT Command; REPEAT Command.
USER	USE	MAP Command.
VALUE	VAL	Match Condition; Unlimited Match Condition.
VLOCATION	VLO	Match Condition; Unlimited Match Condition.
WHILE	W	COUNT Command; REPEAT Command.
WITH	WIT	GO Command; GR Command; TR Command.
WRITE	WRI	WRITE Command.
XADDR	XAD	Trace Reference; Match Condition; Unlimited Match Condition.
XBYTE	XBY	Expression; XBYTE Commands.
XOR	XOR	Expression.
Y	Y	BASE Command; SUFFIX Command

Format:

```
ASM [ ORG address ]
    [ instruction ]
```

Elements:

ASM. Displays or sets the emulator's assembly pointer, or assembles one instruction into code memory at the assembly pointer location. After assembling any instruction, displays the updated assembly pointer.

ORG. Sets the assembly pointer to the *address* specified.

address. Any number, reference, or expression representing a location in code memory. See Address under Expression for details.

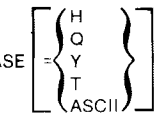
instruction. Any ASM-51 instruction mnemonic except CALL and JMP (the form JMP@A+DPTR is allowed). See The Operating Instructions for details (look in The Index for ASM command).

Examples:

```
*ASM                  ;Display assembly pointer.
*ASM ORG 0100H        ;Set assembly pointer.
*A MOV A,@R0          ;Assemble instruction (note
                      ;abbreviation).
```

BASE Command

Format:



Elements:

BASE. Displays or sets the numeric radix for numbers displayed at the console by the system (except for addresses and certain other types of values noted below.) See also SUFFIX.)

H. Hexadecimal radix; hexadecimal is the initial BASE.

Q. Octal radix; letter "O" may also be used for octal radix.

Y. Binary radix.

T. Decimal radix.

ASCII. ASCII representation of each byte (values 20H through 7EH are displayable characters).

Examples:

```
*BASE           ;Display current output radix.
*BASE = H       ;Set hexadecimal radix.
*BAS = ASC      ;Set ASCII representation.
```

NOTE

The following displays are not affected by the current BASE:

Type of Value	Display Radix
address	Symbolic or hexadecimal.
MAP	Hexadecimal.
BUFFERSIZE	Decimal
REGISTERS	Hexadecimal and binary.
STACK	Hexadecimal
INTERRUPT	Binary

Format:

$$\left\{\begin{matrix}\text{BR0}\\\text{BR1}\\\text{BR}\end{matrix}\right\} = \left[\ \textit{unlimited-match-condition}\ \right]$$

Elements:

BR0. Displays or sets breakpoint register BR0.

BR1. Displays or sets breakpoint register BR1.

BR. Displays both breakpoint registers, or sets both breakpoint registers to the same condition.

unlimited-match-condition. Specifies one or more (non-overlapping) fields of processor information to use for matching, and gives a single value, a masked value, or a partition (range) of values for each field. If a partition is used, it may be adjusted by the system. See Unlimited Match Condition for details.

Examples:

```
*BR0                          ;Display BR0.

*BR                           ;Display BR0 and BR1.

*BR1 = ADDR IS                ;Set BR1 to match any address
  .LOOP TO .END               ;in a range.

*BR0 = VALUE IS 10H           ;Set BR0 to match a combi-
  TO 1FH AND P1 IS            ;nation of operand and port
  20H TO 2FH                  ;values.
```

NOTES

When a breakpoint is enabled, a match can cause real-time emulation to break; see the GO Command and GR Command. The initial setting of both breakpoint registers is all don't-care bits (i.e., match everything). The BR0, BR1, and BR commands do not reset any values previously set in the specified register; only the bits specified in the (unlimited) match condition are affected (see the RESET commands for details on resetting breakpoint registers.)

BUFFERSIZE Command

Format:

```
BUFFERSIZE
```

Elements:

BUFFERSIZE. Displays the number of frames of trace information currently in the trace buffer. BUFFERSIZE is always a decimal number; the values are from 0 to 1000.

Example:

```
*BUFFERSIZE          ;Displays current value.
*BUF
```

CAUSE Command

Format:

```
CAUSE
```

Elements:

CAUSE. Displays the cause of the last break in emulation.

Examples:

```
*CAUSE                ;Displays numeric value and
                      ;text message.

*CAU
```

NOTES

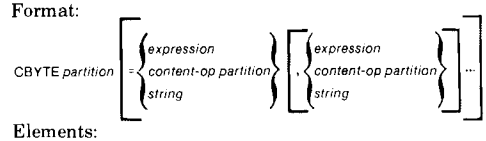
CAUSE produces a 5-bit numeric value, as follows:

Bit 0 = 1 means BR0 match caused break.
Bit 1 = 1 means BR1 match caused break.
Bit 2 = 1 means SY0 caused break.
Bit 3 = 1 means single-stepping caused break.
Bit 4 = 1 means user aborted emulation.

If BR0 and BR1 are both enabled and both happen to match within five addresses, CAUSE may indicate that both caused the break.

CBYTE Commands

Format:



Elements:

CBYTE. Displays or sets the contents of one or more locations in code memory. CBYTE can display from code addresses 0 to 64K; however, CBYTE can write only to the 8K mapped emulator memory.

partition. A single address, or a range of addresses expressed as *address* TO *address* or *address* LENGTH *address.* If the data requires more memory than the size of the partition, an error occurs. If the data requires less memory, the data is repeated until the partition is filled. See Partition.

expression. A number, reference, or formula resulting in a number. See Expression.

content-op. One of the content-operators CBYTE (code memory), DBYTE (on-chip data memory), RBYTE (register memory), XBYTE (external data memory with verification), PBYTE (external data memory without verification), or RBIT (bit-addressable memory). See Expression.

string. One or more characters enclosed in apostrophes ('). The content is the ASCII value of each character, in successive bytes.

Examples:

```
*CBYTE 0100                ;Display contents of one
                           ;address.

*CBYTE .START              ;Display ten consecutive bytes.
  LENGTH 10T

*CBYTE 0 = CBYTE           ;Copy four bytes.
  0100 TO 0103

*CBY 0200H = CBYTE         ;Set 32 bytes
  0105 LEN 27H,
  'ABCDE'

*CBY 100H = "RETI"         ;Using one-byte mnemonic
                           ;constant.
```

Change Commands

A Change command allows you to set or change the value of some element of the emulator system or of the emulation processor. Here is a list of the command entries in this Command Dictionary that contain Change commands:

ASM Command
BASE Command
BR0, BR1, BR Commands
CBYTE Commands
DBYTE Commands
DPTR Command
GO Commands
GR Commands
MAP Command
PBYTE Commands
PC Command
QR0, QR1, QR Commands
RBIT Command
RBS Commnd
RBYTE Commands
R0 - R7 Commands
SUFFIX Command
Symbolic Reference Commands
TM0, TM1 Commands
TR Commands
TRACE Mode Command
XBYTE Commands

COUNT Command

Format:

```
COUNT decimal-expression cr
[ command cr                ]
[ WHILE boolean-expression cr ] ...
[ UNTIL boolean-expression cr ]
  ENDCOUNT
```

Elements:

COUNT. Begins a block of commands to be executed a specified number of times.

decimal-expression. A decimal number, a reference, or an expression in which the default radix for numeric constants is decimal.

cr. An intermediate carriage return. In a block command such as COUNT, all carriage returns are intermediate (terminate the input line but do not terminate the command entry) until the ENDCOUNT keyword is entered.

command. Any emulator command except DEFINE Macro or REMOVE Macro.

WHILE. Specifies that the loop can halt when the condition described in the boolean-expression is tested and found to be FALSE. A halt due to WHILE overrides the count.

UNTIL. Specifies that the loop can halt when the condition in the boolean expression is tested and found to be TRUE. A halt due to UNTIL overrides the count.

boolean-expression. Any of the forms of expression; the condition is TRUE if the least significant bit of the result is 1, FALSE otherwise. See Expression.

ENDCOUNT. Terminates the block of commands.

Examples:

```
*COUNT 10                ;Ten times through the block.
.*GO FROM .START         ;Emulate the program.
  TILL .DONE
.*PRINT -25              ;Display trace.
.*ENDCOUNT               ;Ends the block.

*COU 5                   ;Note abbreviation.
.*STEP                   ;Single step
.*REGISTERS              ;Display registers
.*END                    ;Note abbreviation.
```

DASM Command

Format:

DASM *partition*

Elements:

DASM. Disassembles the contents of the specified partition of user program memory into assembler mnemonics and operands.

partition. A single address, or a range of addresses expressed as *address* TO *address* or *address* LENGTH *address.* (See Partition.) An instruction is displayed if its first byte is within the partition, even if subsequent bytes are outside.

Examples:

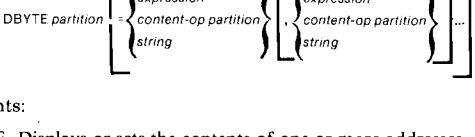
```
*DASM .START              ;Display disassembly for one
                          ;instruction.
*DASM .START TO           ;Partition of addresses
  .DONE
*D 0100H LENGTH 20H       ;Note abbreviation.
```

NOTE

DASM uses symbolic display for addresses in operands and to identify opcode addresses. This display is not affected by DISABLE SYMBOLIC or by BASE.

DBYTE Commands

Format:



Elements:

DBYTE. Displays or sets the contents of one or more addresses in on-chip memory. For ICE-51, DBYTE addresses are 00H to 7FH. For ICE-44, DBYTE addresses are 00H to BFH.

partition. A single address, or a range of addresses expresses as *address* TO *address* or *address* LENGTH *address.* If the data requires more memory than the size of the partition, an error occurs. If the data requires less memory, the data is repeated until the partition is filled. See Partition.

expression. A number, reference, or a formula that evaluates to a number. See Expression.

content-op. One of the content operators CBYTE (code memory), DBYTE (data memory, RBYTE (register memory), XBYTE (external data memory, verified) PBYTE (unverified external data memory), or RBIT (bit-addressable memory). See *Expression.*

string. One or more characters enclosed in apostrophes ('). The content is the ASCII value of each character in successive bytes.

Examples:

```
*DBYTE 0                    ;Display contents of single
                            ;address

*DBYTE 0 TO 7FH = 0         ;Sets entire ICE-51 range
                            ;to zero

*DBYTE 0 TO BFH = 0         ;Sets entire ICE-44 range
                            ;to zero

*DBY 30H = CBYTE            ;Copy to partition.
  .TABLE LENGTH 24T

*DBY 4AH = 'ABCDEF'         ;Note abbreviation.
  ,80H,.COUNT +
  5,XBYTE 07FFH
```

DEFINE Macro Command

Format:

```
DEFINE :macro-name cr
[command cr]...
EM
```

Elements:

DEFINE. Enters user-defined macro name and block of commands into the macro definition table.

:macro-name. The name of the macro block preceded by a colon. The macro name can be 31 characters maximum. The first character must be a letter, @, or ?. The remaining characters may be these characters or numerals.

cr. Intermediate carriage return. In a block such as DEFINE macro, all carriage returns are intermediate (terminate input line but do not terminate the command definition) until the EM ("end of macro") keyword is encountered.

command. Any emulator command except DEFINE Macro and REMOVE Macro. Calls to other macros are allowed, but a macro may not call itself. Commands may contain formal parameters (%0 through %9) to be substituted with actual text parameters at macro invocation (see *Macro Invocation Command*).

EM. Terminates the macro definition.

Examples:

```
*DEFINE :GOER          ;Start macro block.
.*GO FROM .START       ;Emulate the program.
  TILL .DONE
.*PRINT -25            ;Display trace.
.*EM                   ;End of macro definition

*DEF :M                ;Note abbreviation.
.*MAP = %0000H,        ;%0 and %1 are formal text
  %1000H               ;parameters
.*MAP                  ;Display map.
.*EM
```

NOTE

See the Macro Invocation Command for examples of calls to these macros.

DEFINE Symbol Command

Format:

DEFINE *.symbol-name = expression*

Elements:

DEFINE. Enters user-defined symbol and corresponding numeric value into user symbol table.

.symbol-name. The name of the symbol preceded by a period. Symbol names can be 31 characters maximum. The first character must be a letter, @, or ?; the remaining characters may be these characters or numerals. The symbol name may not duplicate either a system symbol or a previously defined user symbol (see System Symbols).

expression. A number, reference, or formula that evaluates to a number. See Expression.

Example:

```
*DEFINE .START =          ;Add new symbol to table.
  0100H

*DEFINE .LOOP = .START ;Note abbreviation.
   + 15H
```

DIR Command

Format:

```
DIR
```

Elements:

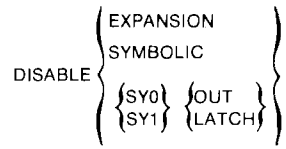
DIR. Displays directory of the names of all macros in the macro definition table, in the order they were defined.

Example:

```
*DIR
```

DISABLE Commands

Format:



Elements:

DISABLE. Cancels the effect of its object on the emulator system.

EXPANSION. When enabled, causes display of macro expansion on the console prior to execution. When disabled, macro expansion is not displayed. (See Macro Invocation Command.)

SYMBOLIC. When enabled, causes display of addresses in symbolic format (*.symbol-name* or *.symbol-name + offset*). When disabled, addresses are displayed in hexadecimal radix.

SY0. External signal SY0, an indicator and control for realtime emulation.

SY1. External signal SY1, an indicator and control for trace collection during real-time emulation.

OUT. When enabled, out causes the emulator to set SY0 or SY1 bigb to indicate to an external device that emulation (SY0 OUT) or trace collection (SY1 OUT) is occurring. When disabled, no external signal is produced.

LATCH. When enabled, LATCH causes system response on SY0 IN or SY1 IN signals to be triggered by a high-to-low transition and to remain latched low until emulation restarts. When disabled, system responses are level-triggered; under this condition SY1 IN can toggle trace on and off, but SY0 IN cannot toggle emulation on and off.

Examples:

```
*DISABLE EXPANSION    ;Macro expansion is initially
                      ;disabled.
*DIS SYMBOLIC         ;Symbolic display is initially
                      ;enabled.
*DIS SY0 OUT          ;OUT applies to the specified
                      ;synch line.
*DIS SY1 LAT          ;Note abbreviations to DIS and
                      ;LAT.
```

Display Commands

A display command produces an output of a value or other message on the console. Here is a list of command entries in this Command Dictionary that contain display commands:

ASM Command
BASE Command
BR0, BR1, BR Commands
BUFFERSIZE Command
CAUSE Command
CBYTE Commands
DASM Command
DBYTE Command
DIR Command
DPTR Command
ERROR Command
EVALUATE Command
HELP Command
HTIMER Command
INTERRUPT Command
MACRO Display Command
MAP Command
OPCODE Command
PBYTE Command
PC Command
PPC Command
PRINT Command
QR0, QR1, QR Commands
RBIT Commands
RBS Command
RBYTE Commands
REGISTER Command
R0 - R7 Commands
SECONDS Command
STACK Command
SUFFIX Command
SY, SY0, SY1 Display Commands
SYMBOLS Command
Symbolic Reference Commands
TIMER Command
TM0, TM1 Command
TRACE Mode Commands
WRITE Command
XBYTE Commands

DPTR Commands

Format:

DPTR [= *expression*]

Elements:

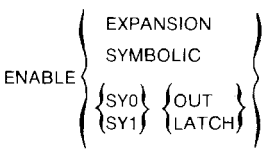
DPTR. Displays or sets the 16-bit Data Pointer Register.

expression. A number, reference, or formula that evaluates to a number. 16 bits are assumed. The low 8 bits are stored in DPL (address 82H in Register memory), and the high 8 bits are stored in DPH (address 83H in Register memory).

Examples:

```
*DPTR                   ;Display data pointer.
*DPTR = 1FA7H           ;Set to 16-bit value.
*DPT = .TABLE + 5       ;Note abbreviation.
```

Format:



Elements:

ENABLE. Causes its object to affect system response.

EXPANSION. Display of macro expansion on the console prior to execution. (When disabled, macro expansion is not displayed.) (See Macro Invocation Command.)

SYMBOLIC. Display of addresses in symbolic format (*.symbol-name* or *.symbol-name + offset*). (When disabled, addresses are displayed in hexadecimal radix.)

SY0. External signal SY0, an indicator and control for realtime emulation.

SY1. External signal SY1, an indicator and control for trace collection during real-time emulation.

OUT. When enabled, OUT causes the emulator to set SY0 or SY1 high to indicate to an external device that emulation (SY0 OUT) or trace collection (SY1 OUT) is occurring. When disabled, no external signal is produced.

LATCH. When enabled, LATCH causes system response on SY0 IN or SY1 IN signals to be triggered by a high-to-low transition and to remain latched low until emulation restarts. When disabled, system responses are level-triggered; under this condition SY1 IN can toggle trace on and off, but SY0 IN cannot toggle emulation on and off.

Examples:

```
*ENABLE EXPANSION      ;Macro expansion is initially
                       ;disabled.
*ENA SYMBOLIC          ;Symbolic display is initially
                       ;enabled.
*ENA SY0 OUT           ;OUT applies to the specific
                       ;sync line.
*ENA SY1 LAT           ;Note abbreviations ENA and
                       ;LAT.
```

ERROR Command

Format:

```
ERROR
```

Elements:

ERROR. Displays the type of emulator hardware error, in those cases where the most recent emulator command involved the emulator hardware (for example, during emulation), and resulted in an error. Displays date code and version number of the software and firmware. This information is provided for use by Intel service personnel.

Example:

```
*ERROR
```

ERROR MESSAGES AND WARNINGS

Here is a list of the error messages and warnings in hexadecimal order, including brief discussions of each error and suggestions for corrective action.

ERR 12:PAR BLK DVC CODE ERROR
ERR 13:PAR BLK FORMAT ERROR
ERR 14:NON-ZERO COMMAND ACK
ERR 15:NON-ZERO DONE FLAG
ERR 16:DVC CD FORMAT ERROR
ERR 17:DVC NOT IN DVC CD TABLE

Errors 12 - 17: Hardware/software communication failure. (1) RESET ICE. (2) Check emulator hardware installation. (3) Run the confidence test. (4) Verify development system hardware operation.

ERR 20:ILLEGAL INPUT COMMAND

The emulator software is not giving a correct command to the hardware. (1) EXIT, then invoke the emulator again. (2) Substitute a fresh software diskette. (3) Run the confidence test.

ERR 21:COMMAND NOT ALLOWED NOW
ERR 22:RSLTS LENGTH INADEQUATE
ERR 23:COMMAND FORMAT ERROR

Errors 21, 22, and 23: Software/hardware communication problem; probable software error. (1) RESET ICE. (2) EXIT, then invoke the emulator again. (3) Substitute a fresh software diskette. (4) Run the confidence test.

ERR 30:PGM MEMORY FAILURE

After command to change user program memory (in buffer box), the data read back did not agree with the data written. (1) RESET ICE. (2) EXIT and run the confidence test.

ERR 31:DATA MEMORY FAILURE

After command that changed external data memory, data read back did not agree with data written. (1) RESET ICE. (2) Check user system.

ERR 38:CONTROL MEMORY FAILURE

During hardware reset, data read back from controller memory did not agree with data written. (1) EXIT and invoke the emulator again. (2) Run the confidence test.

ERR 39:FIRMWARE CHECKSUM ERR

Checksum error after downloading firmware from diskette file ICE51.OVS. (1) EXIT and invoke the emulator again. (2) Substitute a fresh software diskette. (3) Run the confidence test.

ERR 40:NO USER CLOCK

No clock signal at the emulator plug. Ensure that plug is installed correctly and (when connected to user system) that correct crystal is present. (1) RESET ICE. (2) Check the user hardware. (3) Run the confidence test.

ERR 41:NO USER VCC

No power or temporary loss of power at the emulator plug. Ensure that plug is installed correctly and (when connected to user system) that power is present. (1) RESET ICE. (2) Check the user hardware. (3) Run the confidence test.

ERR 43:PROCESSOR NOT RUNNING

The EA pin is inactive although no emulator program memory is mapped to the first 4K block, or the RESET pin is being held high while a command is being processed (i.e., not in emulation and no prompt is displayed). (1) Check the user system operation of these two pins. (2) RESET ICE. (3) Run the confidence test.

ERR 51:BANK SWITCHING HUNG

The emulator has lost synchronization between the emulator hardware and the 8051 processor. (1) RESET ICE. (2) EXIT and invoke the emulator again. (3) Check the user system for EA or RESET changing during command processing. (4) Run the confidence test.

ERR 52:UNWRITEABLE MEMORY

The user attempted to write to external program memory.

ERR 80:SYNTAX ERROR

The token flagged by a #-sign in the previous line is not allowed in this command context. The command is ignored.

ERR 81:INVALID TOKEN

The token flagged hy a #-sign in the previous line is not properly formed, or is inappropriate for the command context; for example, the entry CBYTE 300 when SUFFIX = Y (binary).

ERR 83:INAPPROPRIATE NUMBER

Inappropriate value for the current context; value outside the range permitted for the specified memory or register type.

ERR 84:PARTITION BOUNDS ERROR

Incorrect values used to define a memory partition. Left bound exceeds right bound, or value out of range in the current context.

ERR 85:ITEM ALREADY EXISTS

DEFINE command refers to a symbol or macro already defined.

ERR 86:ITEM DOES NOT EXIST

Symbol or macro in command is not defined. (1) For a symbol, check version of code loaded, or use DEFINE Symbol. (2) For a macro, define or INCLUDE the macro definition.

ERR 88:MACRO PARAMETER ERROR

A macro call contained more than ten actual parameters, or a parameter contained too many characters. (1) Check macro definition.

ERR 89:MISSING CR-LF IN FILE

The current INCLUDE file does not end with a carriage return/linefeed. (1) EXIT and use your editor to correct the file.

ERR 8E:TRACE FRAME EMPTY

A trace reference (e.g., FRAME ADDR) has been used in an expression, but either no trace has been collected in the buffer or the buffer pointer is at NEWEST. (1) Check the contents of the trace buffer (PRINT command), (2) Move the pointer to the desired frame (MOVE command).

ERR 8F:NON-NULL STRING NEEDED

A null string (apostrophes with no enclosed characters) was used where at least one character is required, such as DBYTE 1 = ' '.

ERR 90:MEMORY OVERFLOW

The emulator workspace (for symbols and macro expansion) exceeded the amount allocated to it. (1) To reclaim workspace, remove some user symbols.

ERR 92:COMMAND TOO LONG

The command exceeds the capacity of the emulator's command buffer. Possibly caused by too many operators. (1) Break the command or expression into several smaller units.

ERR 94:NON-CHANGEABLE ITEM

The command attempted to change a read-only value such as PPC or BUFFERSIZE.

ERR 95:INVALID OBJECT FILE

The hexadecimal object file referenced in a LOAD command is not written in the proper format. (1) Select another file for loading. (2) EXIT and verify file type.

ERR 99:EXCESSIVE ITERATED DATA

The number of data items to be repeated in memory exceeds the buffer size for iterated data (128 bytes). Example: CBYTE 30 TO 1000H = RBYTE 0 TO 256T.

ERR 9D:LINE TOO LONG

The input line exceeds 120 characters. (1) Use continuation lines. (2) Break command into two or more shorter commands.

ERR A4:MACRO FILE FULL

The temporary file MAC.TMP is full. (1) Save and remove macro definitions to make room for more, using the PUT and REMOVE MACRO commands.

ERR A9:MAP CONTENTS CHANGED

Data read back from the map does not agree with data previously written. (1) Set and display the map again. (2) Run the confidence test.

ERR B3:OFFSET TOO LARGE

An address in an 8051 instruction (for ASM or as a mnemonic constant) results in a relative offset that is larger than 8 bits. (1) Check the relative offset; you may need a different instruction type.

ERR B7:PARTITIONS NOT ALLOWED

Command specified a range of values for a match condition in a GO command, GR command, or TR command. Partitions for match conditions are allowed only in the BR and QR commands.

ERR B8:ASSEMBLY IMPOSSIBLE

A mnemonic instruction in an ASM command or mnemonic constant has been specified incorrectly. (1) Check the correct instruction format and enter the instruction again.

ERR B9:NO HELP AVAILABLE

No HELP message is available for the item requested. (1) Type HELP with no modifier to obtain a menu of available HELP items.

ERR BB:ILLEGAL MAP BOUNDARY.

The map boundary in the previous MAP command does not fall on a 4K boundary. Map blocks must begin at multiples of 1000H (4K).

ERR BC:SYSTEM SYMBOL ERROR

Invalid operation on system symbol table, such as attempting to change the value of a system symbol, or using a multiple reference (e.g., .P0.P1) involving system symbols.

ERR BD:INVALID REG BANK NUMBER

The command attempted to set the register bank select (RBS) to a value other than 0, 1, 2, or 3.

WARN CO:EXCESSIVE DATA

Partition of memory too small for the number of data items specified. For example, attempting to load 50 bytes into a partition only 40 bytes long; in this case, the first 40 bytes are loaded correctly but the last 10 bytes of the data are ignored.

WARN C1:80XX CHIP FAILURE

The system has tested the chip and discovered it is not the one specified by the user's invocation. For instance, the "ICE-51" message was entered when the chip being used is an 8044. Retry the invocation.

WARN C2:HARDWARE MISSING

The device with the device code on the previous line did not respond to initialization. (1) Check that controller board is set to device code 32H and that all boards are correctly installed (refer to appendix A for details).

WARN C3:MULTIPLE HARDWARE

Two or more emulators are installed with the same device codes. (1) Check installation and device code setting so that only one ICE emulator (device code 32H) can respond to initialization.

WARN C4:EXTRA FRAMES

During trace display in INSTRUCTION mode, the system detected extra frames after the end of the instruction. The extra frames cannot be assigned to any instruction. Probable cause is that the trace register setting qualified trace collection so as to omit some LOCATION frames.

WARN C9:VERSIONS DO NOT MATCH

Version numbers of software and downloadable firmware are not correct. (1) Verify that files ICE51 or ICE44 and ICE51.OVS or ICE44.OVS are present on software diskette. (2) Substitute a fresh software diskette.

WARN CA:PPC/OPCODE NOT VALID

In updating trace information after emulation, up to 16 of the newest frames are searched for a LOC frame to refresh the values of PPC and OPCDE. If a LOC frame is not found in those 16 frames, PPC and OPCODE will contain invalid data.

WARN CB:TRUNCATED TO 8 BITS

OPCODE, VALUE, or port value in a match condition, or an expression in a mnemonic instruction to be assembled, is greater than 8 bits. The value is truncated to the low 8 bits.

WARN CC:UNEXPECTED TRACE

During trace update after breaking emulation or during trace display, code memory does not match the data collected in trace.

WARN CD:TRUNCATED TO 11 BITS

An expression value (in an ACALL or AJMP instruction for assembly) is larger than 11 bits. The value is truncated to 11 bits.

ERR E7:ILLEGAL FILENAME

The command specified a filename that does not conform to ISIS-II format specifications. (1) Check the ISIS-II User's Guide.

ERR E8:ILLEGAL DEVICE

The command contained illegal or unrecognizable reference to an ISIS-II device. (1) Check the ISIS-II User's Guide.

ERR E9:FILE OPEN FOR INPUT

Command attempted to write data to a file opened for input (read-only) operations. (1) Select another file for writing. (2) Close file and reopen.

ERR EF:FILE ALREADY OPEN

Command attempted to open a file that is already open.

ERR F0:NO SUCH FILE

Command specified file that does not exist on the designated diskette. (1) Verify drive number, filename, and diskette.

ERR F1:WRITE-PROTECTED FILE

The command attempted to open a write-protected file for write access. (1) Select another file. (2) EXIT and remove write protection.

ERR F3:CHECKSUM ERROR

Checksum error in hexadecimal object file during loading. (1) Select another file. (2) EXIT and create a correct object file.

ERR F6:DISKETTE FILE REQUIRED

The command specified a device other than a diskette file, but the operation requires a diskette file.

ERR F9:ILLEGAL ACCESS

Command attempted to open a read-only device for write access, or attempted to open a write-only device for read access. (1) Check command syntax for list of appropriate devices.

ERR FA:NO FILE NAME

The command references a diskette device, but omitted the filename.

ERR FD:"DONE" TIMED OUT
ERR FE:"ACKNOWLEDGE" TIMED OUT

The emulator was unable to complete a requested operation. (1) Check hardware installation. (2) RESET ICE. (3) EXIT and invoke emulator again. (4) Run the confidence test.

ERR FF:NULL FILE EXTENSION

The command referenced a file terminated by a period, but the implied extension is missing. (1) Omit the period or include the extension (refer to the ISIS-II User's Guide).

EVALUATE Command

Format:

EVALUATE *expression*

Elements:

EVALUATE. Performs the specified mathematical operations to evaluate the expression to a (16-bit) numeric result, then displays the result in binary radix, octal radix, decimal radix, and hexadecimal radix; as a pair of ASCII characters (for byte values from 20H through 7EH) enclosed in apostrophes; and as an address. The address displayed is the name of the user symbol that is closest to but less than or equal to the result, plus the offset if any; if no user symbol meets this test, the address appears in hexadecimal.

expression. A number, reference, or a formula that can be evaluated to a numeric result. See Expression.

Examples:

```
*EVALUATE 10T + 2AH     ;Perform addition and display
                        ;result.
*EVA (RBYTE .ACC)       ;Note abbreviation to EVA.
  + DPTR
```

EXIT Command

Format:

```
EXIT
```

Elements:

EXIT. Terminates the emulator session; closes all LIST files (if any); removes the temporary macro definition file; returns control to the ISIS-II system.

Examples:

```
*EXIT
*EXI
```

Expression

This section presents a formal definition of the term *expression* as it appears in emulator command formats, and in addition defines two special kinds of expression: address and boolean expression.

The term *expression* in a command format means that all of the operands and operators described below are allowed. Typically, an expression evaluates to a 16-bit value; depending on the context, the result may be truncated to fewer than 16 bits.

An expression has the general format:

[*unary-op*] ... *operand* [*binary-op* [*unary-op*] ... *operand*]...

The elements of expression are:

unary-op. A unary operator takes one operand. Unary operators are:

+	Unary plus.
-	Unary minus (two's complement)
CBYTE	Treats its operand as a code address, returns the contents (byte) of that address.
DBYTE	Contents of data memory.
PBYTE	Contents of external data memory, without verification.
RBYTE	Contents of register memory.
XBYTE	Contents of external data memory, with verification.
RBIT	Bit content of bit-addressable memory.
NOT	One's complement.

binary-op. A binary operator takes two operands. Binary operators are:

*	Multiplication.
/	Integer division.
MOD	Modulo reduction (remainder after division).
+	Addition.
-	Subtraction.
=	Equals (result is TRUE (FFFFH) if the two operands are equal, FALSE (0000H) otherwise).
>	Greater than (TRUE or FALSE relation).
<	Less than (TRUE or FALSE relation).
>=	Greater than or equal to (TRUE or FALSE relation).
<=	Less than or equal to (TRUE or FALSE relation).
<>	Not equal to (TRUE or FALSE relation).
AND	Bitwise logical AND.
OR	Bitwise logical OR.
XOR	Bitwise logical exclusive OR.

Precedence of operators determines when parentheses are required to avoid syntax errors. In the following list, 1 is the highest precedence (evaluated first), and 8 is the lowest:

Precedence	Operators
1	+, - (unary)
2	*, /, MOD
3	+, - (binary)
4	CBYTE, DBYTE, PBYTE, RBYTE, XBYTE, RBIT
5	=, >, <, >=, <=, <>
6	NOT
7	AND
8	OR, XOR

Evaluation is left to right for operators of equal precedence.

operand. A numeric constant, mnemonic constant, string constant, symbolic reference, keyword reference, trace reference, or an expression enclosed in parentheses. The types of constants and references are defined as follows:

numeric-constant. A number with the general format:

digit [*digit*]...[*radix*]

digit. A numeral (0 to 1 for binary, 0 - 7 for octal, 0 - 9 for decimal and hexadecimal) or a letter (A - F for hexadecimal only).

radix. A letter (Y for binary, Q or O for octal, T for decimal, H for hexadecimal).

mnemonic-constant. An *instruction* enclosed in quotation marks ("). The operand value is the one-byte opcode. See Instruction.

string-constant. A character enclosed in apostrophes ('). The operand value is the ASCII value of the character.

symbolic reference. The name of a user symbol or system symbol preceded by a period. For user symbols, multiple references are allowed. The operand value is the address or other value corresponding to the symbol. See Symbolic Reference Commands.

keyword-reference. One of the processor references or emulator references listed below. The operand value of a keyword reference is the content or status of the designated register or bit.

processor-reference. DPTR, PC, R0, R1, R2, R3, R4, R5, R6, R7, RBS, TM0, TM1.

emulator reference. BUFFERSIZE, CAUSE, HTIMER, OPCODE, PPC, SY0, SY1, TIMER.

trace-reference. The value of a specified field of trace information in the frame given by the trace display pointer. See Trace Reference.

Here are definitions of particular kinds of expressions required by certain command contexts.

Address

Format:

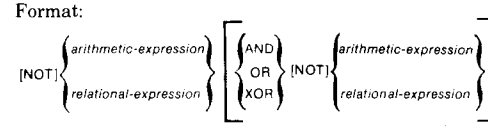
[-] *operand* [*arithmetic-op* [-] *operand*] ...

arithmetic-op: Binary operators (*, /, MOD, +, -)

In an *address*, only arithmetic operators are allowed outside parentheses.

Boolean Expression

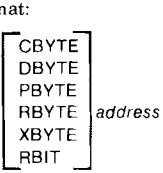
Format:



As an entry, *boolean-expression* is identical to *expression.* As a numeric context, however, the result of a boolean expression is either TRUE (1) or FALSE (0); only the least significant bit of the result is tested.

Arithmetic Expression

Format:



Relational Expression

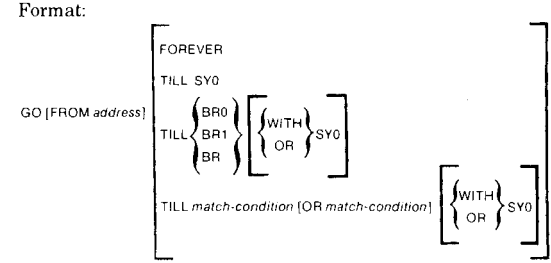
Format:

arithmetic-expression relational-op arithmetic-expression

relational-op: One of the relational operators (=, >, <, >=, <=, <>).

The result of a relational expression is TRUE (FFFFH) if the designated relation holds at the time the expression is evaluated, FALSE (0000H) otherwise.

Format:



Elements:

GO. Begins real-time emulation immediately or (if SY0 IN is enabled) when SY0 goes high.

FROM. Specifies the start address for emulation. Changes the program counter prior to the start of emulation; resets the emulation timer and the trace buffer. If the FROM clause is omitted, the current PC is used.

address. A number, reference, or expression (see Expression).

FOREVER. Disables all halt factors for real-time emulation except user abort (ESC Key).

TILL SY0. Enables external signal SY0 IN to halt emulation by going low, or to control the start of emulation by being low when the GO command is entered. (See ENABLE Commands for the Latch/Level control of SY0 IN.)

TILL BR0, TILL BR1, TILL BR. Enables the designated breakpoint register (both breakpoint registers with BR) to halt real-time emulation by matching the current frame of processor data.

WITH SY0. Specifies that the designated breakpoint register match and SY0 IN low must be true simultaneously to halt emulation.

OR SY0. Specifies that either the designated breakpoint match or SY0 IN going low can halt emulation.

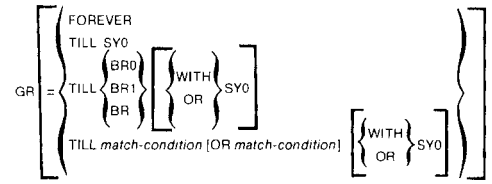
TILL *match-condition*. Sets the match condition in BR0 and enables BR0 to halt emulation by matching. See Match Condition.

TILL *match-condition* OR *match-condition.* Sets the first match condition in BR0; sets the second match condition in BR1; enables both BR0 and BR1. See Match Condition.

Examples:

```
*GO                          ;Accepts default PC and halt
                             ;conditions.

*GO FOREVER

*GO FROM .START              ;Note abbreviation FOR
  FOR                        ;(FOREVER).

*G F 0100 T BR0              ;Note abbreviations G, F and T.

*G T LOCATION IS             ;Uses two breakpoint registers.
  0100 OR VALUE IS
  F7H WITH SY0
```

Format:



Elements:

GR. Displays or sets halt conditions for real-time emulation without starting emulation.

FOREVER. Disables all halt factors for real-time emulation except user abort (ESC key).

TILL SY0. Enables external signal SY0 IN to halt emulation by going low, or to control the start of emulation by being low when the GO command is entered. (See ENABLE Commands for the Latch/Lever control of SY0 IN).

TILL BR0, TILL BR1, TILL BR. Enables the designated breakpoint register (both breakpoint registers with BR) to halt real-time emulation by matching the current frame of processor data.

WITH SY0. Specifies that the designated breakpoint register match and SY0 IN low must be true simultaneously to halt emulation.

OR SY0. Specifies that either the designated breakpoint match or SY0 IN going low can halt emulation.

TILL *match-condition*. Sets the match condition in BR0 and enables BR0 to halt emulation by matching. See Match Condition.

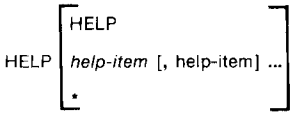
TILL *match-condition* OR *match-condition*. Sets the first match condition in BR0; sets the second match condition in BR1; enables both BR0 and BR1. See Match Condition.

Examples:

```
*GR                          ;Displays the current halt
                             ;conditions.
*GR = FOREVER                ;Disables all halt factors
*GR = TILL SY0
*GR = T LOCATION             ;Note abbreviation T for TILL
  IS 0100H
```

HELP Command

Format:



Elements:

HELP. Displays a menu of available help items; displays explanation of one or more help items.

HELP HELP. Displays explanation of the HELP command and of the notation used in explaining help items. (Because it can use the screen display characters only, the HELP syntax notation differs somewhat from the command format notation used in this manual.)

help-item. An emulator command or command term for which an explanation is available through the HELP command.

HELP *. Displays the explanations of all items in the help file (useful for obtaining a hard-copy listing).

Examples:

```
*HELP                ;Display HELP items available.
*HELP GO             ;Help with the GO command.
*HELP GR, TR         ;Help with both GR and TR
                     ;commands.
*HELP *              ;Display all HELP messages.
```

HTIMER Command

Format:

```
HTIMER
```

Elements:

HTIMER. Displays the most significant 16 bits of the 32-bit half-microsecond emulation timer. The emulation timer is cleared whenever the program counter changes, or during single-step emulation. (See TIMER Command, SECONDS Command).

Example:

```
*HTIMER
```

IF Command

Format:

```
IF boolean-expression [THEN] cr
[command cr] ...
[ORIF boolean-expression cr] ...
[command cr] ...
[ELSE cr]
[command cr] ...]
ENDIF
```

Elements:

IF. Begins a block of commands to be executed when the IF condition is TRUE.

boolean-expression. Any of the forms of expression. When the boolean expression after IF or ORIF is TRUE, the commands in that block are executed to the exclusion of all other blocks in that IF command.

THEN. Optional addition to the IF command.

cr. Intermediate carriage return; ends each input line of the IF command, but does not terminate the entry of the command.

command. Any emulator command except DEFINE Macro and REMOVE Macro.

ORIF. Introduces a secondary block of commands. When the condition after the IF is FALSE, the first ORIF condition found to be TRUE causes that ORIF block to be executed to the exclusion of all other command blocks.

ELSE. Introduces a block of commands to be executed when no IF or ORIF condition is TRUE.

ENDIF. Terminates the entire IF command.

Example:

```
*IF PC = .RESET
.*GO FROM .START          ;Executed only if PC = 0
  TILL .DONE              ;(.RESET)
.*ORIF PC = .EXTIO        ;Executed only if PC not 0.
.*GO FROM .IO TILL        ;Executed only if IF not true
  OPCODE IS "RETI"        ;and PC = 03H
.*ELSE                    ;Executed only when all other
.*GO FROM .I1 TILL SY0    ;conditions are false.
.*ENDIF                   ;Terminates IF command block
```

INCLUDE Command

Format:

INCLUDE :F*n:filename*

Elements:

INCLUDE. Reads a sequence of emulator commands from the indicated file and drive. Each command is executed as it is read in (if it is an executable command), then the emulator system prompts for the next command. One use for INCLUDE is to retrieve macro definitions saved with the PUT command.

:F*n:filename*. The drive number (n = 0 - 9), and the name of the file containing the command sequence.

Example:

*INCLUDE :F1:TEST5.MAC

Instruction

The following tables list the 8051 instructions in functional groups and gives the number of bytes, the number of cycles, and the maximum number of trace frames required for each instruction. These instruction formats are valid for entry using the emulator ASM command.

The notational form @R*n* in some instruction formats means @R0 or @R1; the notational form R*n* means R0, R1, R2, R3, R4, R5, R6, or R7.

Arithmetic Instructions

Instruction		Opcode	Bytes	Cycles	Frames
ADD	A,*data-address*	25H	2	1	4
ADD	A,R*n*	28H-2FH	1	1	4
ADD	A,@R*n*	26H-27H	1	1	4
ADD	A,#*data*	24H	2	1	4
ADDC	A,*data-address*	35H	2	1	4
ADDC	A,R*n*	38H-3FH	1	1	4
ADDC	A,@R*n*	36H-37H	1	1	4
ADDC	A,#*data*	34H	2	1	4
CLR	A	E4H	1	1	4
CLR	*bit-address*	C2H	2	1	4
CLR	C	C3H	1	1	4
CPL	A	F4H	1	1	4
CPL	*bit-address*	B2H	2	1	4
CPL	C	B3H	1	1	4
DA	A	D4H	1	1	4
DEC	A	14H	1	1	4
DEC	*data-address*	15H	2	1	4
DEC	R*n*	18H-1FH	1	1	4
DEC	@R*n*	16H-17H	1	1	4
DIV	AB	84H	1	4	16
INC	A	04H	1	1	4
INC	*data-address*	05H	2	1	4
INC	DPTR	A3H	1	2	8
INC	R*n*	08H-0FH	1	1	4
INC	@R*n*	06H-07H	1	1	4
MUL	AB	A4H	1	4	16
SETB	*bit-address*	D2H	2	1	4
SUBB	A,*data-address*	95H	2	1	4
SUBB	A,R*n*	98-9FH	1	1	4
SUBB	A,@R*n*	96H-97H	1	1	4
SUBB	A,#*data*	94H	2	1	4

Control Transfer Instructions

Instructions		Opcode	Bytes	Cycles	Frames
ACALL	*code-address*	11H	2	2	8
ACALL	*code-address*	31H	2	2	8
ACALL	*code-address*	51H	2	2	8
ACALL	*code-address*	71H	2	2	8
ACALL	*code-address*	91H	2	2	8
ACALL	*code-address*	B1H	2	2	8
ACALL	*code-address*	D1H	2	2	8
ACALL	*code-address*	F1H	2	2	8

Instruction		Opcode	Bytes	Cycles	Frames
AJMP	*code-address*	01H	2	2	8
AJMP	*code-address*	21H	2	2	8
AJMP	*code-address*	41H	2	2	8
AJMP	*code-address*	61H	2	2	8
AJMP	*code-address*	81H	2	2	8
AJMP	*code-address*	A1H	2	2	8
AJMP	*code-address*	C1H	2	2	8
AJMP	*code-address*	E1H	2	2	8
CJNE	A,*data-address,code-address*	B5H	3	2	8
CJNE	A,#*data,code-address*	B4H	3	2	8
CJNE	R*n*,#*data,code-address*	B8H-BFH	3	2	8
CJNE	@R*n*,#*data,code-address*	B6H-B7H	3	2	8
DJNZ	*data-address,code-address*	D5H	3	2	8
DJNZ	R*n,code-address*	D8H-DFH	2	2	8
JB	*bit-address,code-address*	20H	3	2	8
JBC	*bit-address,code-address*	10H	3	2	8
JC	*code-address*	40H	2	2	8
JMP	@A+DPTR	73H	1	2	8
JNB	*bit-address,code-address*	30H	3	2	8
JNC	*code-address*	50H	2	2	8
JNZ	*code-address*	70H	2	2	8
JZ	*code-address*	60H	2	2	8
LCALL	*code-address*	12H	3	2	8
LJMP	*code-address*	02H	3	2	8
RET		22H	1	2	8
RETI		32H	1	2	8
SJMP	*code-address*	80H	2	2	8

Data Transfer Instructions

Instruction		Opcode	Bytes	Cycles	Frames
MOV	A,*data-address*	E5H	2	1	4
MOV	A,R*n*	E8H-EFH	1	1	4
MOV	A,@R*n*	E6H-E7H	1	1	4
MOV	A,#*data*	74H	2	1	4
MOV	*bit-address*,C	92H	2	1	4
MOV	C,*bit-address*	A2H	2	1	4
MOV	*data-address*,A	F5H	2	1	4
MOV	*data-address,data-address*	85H	3	2	8
MOV	*data-address*,R*n*	88H-8FH	2	2	8
MOV	*data-address*,@R*n*	86H-87H	2	2	8
MOV	*data-address*,#*data*	75H	3	2	8
MOV	DPTR,#*data*	90H	3	2	8
MOV	R*n*,A	F8H-FFH	1	1	4
MOV	R*n,data-address*	A8-AFH	2	2	8
MOV	R*n*,#*data*	78H-7FH	2	1	4
MOV	@R*n*,A	F6H-F7H	1	1	4
MOV	@R*n,data-address*	A6H-A7H	2	2	8
MOV	@R*n*,#*data*	76H-77H	2	1	4
MOVC	A,@A+DPTR	93H	1	2	4
MOVC	A,@A+PC	83H	1	2	8
MOVX	A,@DPTR	E0H	1	2	8
MOVX	A,@R*n*	E2H-E3H	1	2	8
MOVX	@DPTR,A	F0H	1	2	7
MOVX	@R*n*,A	F2H-F3H	1	2	7
POP	*data-address*	D0H	2	2	8
PUSH	*data-address*	C0H	2	2	8
XCH	A,*data-address*	C5H	2	1	4
XCH	A,R*n*	C8H-CFH	1	1	4
XCH	A,@R*n*	C6H-C7H	1	1	4
XCHD	A,@R*n*	D6H-D7H	1	1	4

Logical Instructions

Instruction		Opcode	Bytes	Cycles	Frames
ANL	A,*data-address*	55H	2	1	4
ANL	A,R*n*	58H-5FH	1	1	4
ANL	A,@R*n*	56H-57H	1	1	4
ANL	A,#*data*	54H	2	1	4
ANL	C,*bit-address*	82H	2	1	4
ANL	C,/*bit-address*	B0H	2	2	8
ANL	*data-address*,A	52H	2	1	4
ANL	*data-address*,#*data*	53H	3	2	8
CLR	C	C3H	1	1	4
CPL	A	F4H	1	1	4
CPL	C	B3H	1	1	4
NOP		00H	1	1	4
ORL	A,*data-address*	45H	2	1	4
ORL	A,R*n*	48H-4FH	1	1	4
ORL	A,@R*n*	46H-47H	1	1	4
ORL	A,#*data*	44H	2	1	4
ORL	C,*bit-address*	72H	2	1	4
ORL	C,/*bit-address*	A0H	2	2	8
ORL	*data-address*,A	42H	2	1	4
ORL	*data-address*,#*data*	43H	3	2	8
RL	A	23H	1	1	4
RLC	A	33H	1	1	4
RR	A	03H	1	1	4
RRC	A	13H	1	1	4
SETB	*bit-address*	D2H	1	1	4
SETB	C	D3H	1	1	4
SWAP	A	C4H	1	1	4
XRL	A,*data-address*	65H	2	1	4
XRL	A,R*n*	68H-6FH	1	1	4
XRL	A,@R*n*	66H-67H	1	1	4
XRL	A,#*data*	64H	2	1	4
XRL	*data-address*,A	62H	2	1	4
XRL	*data-address*,#*data*	63H	3	2	8

NOTE

In the ACALL and AJMP instructions, the high-order 3 bits of the opcode contain the high-order 3 bits of the *code-address*. Thus the opcode determines the range of the jump or call as follows:

Opcode		Range
AJMP	ACALL	
01H	11H	000H — 0FFH
21H	31H	100H — 1FFH
41H	51H	200H — 2FFH
61H	71H	300H — 3FFH
81H	91H	400H — 4FFH
A1H	B1H	500H — 5FFH
C1H	D1H	600H — 6FFH
E1H	F1H	700H — 7FFH

INTERRUPT Command

Format:

```
INTERRUPT
```

Elements:

INTERRUPT. Displays the Interrupt Priority (IP) register, the Interrupt Enabled (IE) register, the Priority 0 Interrupt In Progress (IIP0) register, and the Priority 1 Interrupts In Progress (IIP1) register, for the five 8051/8044 interrupts TIMER0, TIMER1, EXTI1, and SERINT. (NOTE: the IIP register cannot be set directly through the emulator, although it can be reset; see the RESET CHIP Command.)

Example:

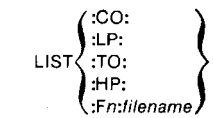
```
*INTERRUPT

*INT                    ;Note abbreviation
```

Lights

Buffer box lights indicate emulator status as described in the following table:

Lights On	Emulator Status	User Action
Yellow	Interrogate Mode; no command in progress.	Enter any command.
None	Command in progress.	To abort command, press ESC key.
Green and Yellow	Emulation in progress.	To abort emulation, press ESC key.
Green	GO Command entered while SY0 IN enabled, and SY0 has never been high; waiting for SY0 to go high so emulation can begin.	To abort command, press ESC key.
Red and Yellow	Hardware error, recovery has occurred. The red-yellow combination occurs momentarily during RESET ICE.	Enter any command.
Red, or Red and Green, or All three	Hardware error, no recovery has occurred.	Try ESC, then RESET ICE, or reset development system.

Format:



Elements:

LIST. Outputs a copy of the emulation session to an external device other than the console (and to the console also).

:CO: Console output only (:CO: is the initial LIST device).

:LP: Line printer.

:TO: Terminal printer or display.

:HP: High-speed punch.

:F*n*:*filename* ISIS-II diskette file. :F*n*: identifies the drive number (*n* = 0 - 9). If the filename does not already exist on the target drive, the system opens the file for input.

Examples:

```
*LIST :F1:JUL21.LOG
*LIS :LP:                ;Note abbreviation.
*LIS :CO:                ;Restores console only after
                         ;using another device.
```

LOAD Command

Format:

LOAD :F*n:filename* [NOCODE] [NOSYMBOLS]

Elements:

LOAD. Loads the user program and symbol table from an object file on diskette into code memory as mapped. When the object file contains a symbol table, the LOAD command loads the table but does not check the symbols for duplications with user symbols already in the user symbol table. Thus, multiple loads of the same program can result in multiple versions of the same user symbol. However, if the program symbol table contains references to system symbols (as allowed by ASM-51), the system symbols are not loaded; (see System Symbols for a list of these symbols).

:F*n:filename*. The drive (*n* = 0 - 9) and name of the file containing the program to be loaded.

NOCODE. Suppresses loading of the code portion of the object file (loads symbol table only, if present).

NOSYMBOLS. Suppresses loading of the symbol table portion of the object file (loads code only).

Examples:

```
*LOAD :F1:PROG.HEX      ;Load code and symbols.
*LOA :F1:PROG.HEX
  NOCODE                ;Note abbreviation.
*LOAD :F1:PROG.HEX
  NOSYMBOLS
```

MACRO Display Command

Format:

MACRO [:*macro-name* [, :*macro-name*] ...]

Elements:

MACRO. Displays the definitions of the macros in the list of macro-names, or (if the list is omitted), displays the definitions of all macros currently in the macro table.

:*macro-name*. The name of a macro as defined previously by the user. The colon (:) identifies the entry as a macro name.

Example:

```
*MACRO                  ;Display enter macro table.
*MAC :GOER              ;Note abbreviation to MAC
*MAC :M, :RPT           ;Display several definitions.
```

Macro Invocation Command

Format:

:macro-name [*actual-parameter* [, *actual-parameter*] ...]

Elements:

:macro-name. Invokes a previously defined macro (sequence of commands). At invocation, the macro definition is expanded: actual parameters in the invocation (if any) are substituted for formal parameters in the macro definition. After expansion, if only valid command syntax has been produced, the commands are executed. (Macro expansion can be displayed if desired; see ENABLE EXPANSION for details.)

actual-parameter. A sequence of characters, tokens, or commands, or a string enclosed in apostrophes (use a string when a single parameter contains embedded commas, to distinguish it from a list of several parameters). The text in each actual parameter is substituted literally for the corresponding formal parameter in the macro definition. (Refer to the *Operating Instructions*, chapter 7, for details on formal and actual parameters).

Examples (see DEFINE Macro Command for the definitions):

```
*:GOER          ;No parameters required.
*:M 0,5         ;Two parameters required.
```

Format:

$$\text{MAP}\left[=\left\{\begin{array}{l}\textit{block-address, block-address}\\ \text{USER}\end{array}\right\}\right]$$

Elements:

MAP. Displays or sets the mapping of the two 4K blocks of code memory supplied by the emulator. Any code memory blocks (in the 64K maximum range) not mapped to the emulator are assumed to be user-supplied.

block-address. The lowest address in a 4K map block. A map block address can be entered as a four-digit multiple of 1000H, or as a decimal multiple of 4K. Two block addresses must be specified in the MAP command.

USER. Specifies that all program memory is to be external, user-supplied memory.

Examples:

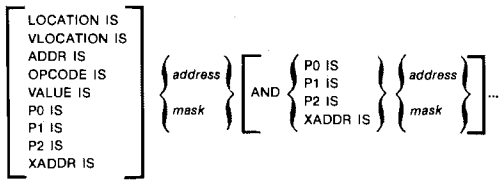
```
*MAP                    ;Displays the current mapping.
*MAP=0,1000H            ;The initial map setting.
*MAP=0, 8K
```

NOTE

When the EA/ pin is inactive (high), mapping the first 4K block to external program memory is not allowed. When the EA/ pin is active, memory should be mapped to blocks other than the first block; external program memory is assumed to exist for this block. If the entire program memory space is external user memory, enter MAP = USER.

Match Condition

Format:



Elements:

LOCATION IS. Specifies that the address or mask value is to be used to match the code address field on LOCATION frames only (address of opcode). LOCATION IS is the default if no frame type is designated in the match condition.

VLOCATION IS. Matches the code address field on VLOCATION (operand address) frames only.

ADDR IS. Matches the code address field on both LOCATION and VLOCATION frames.

OPCODE IS. Matches the data field (low byte of address field) on OPCODE frames only.

VALUE IS. Matches the data field (low byte of address field) on VALUE frames (operand value) only.

P0 IS. Matches the P0 field.

P1 IS. Matches the P1 field.

P2 IS. Matches the P2 field.

XADDR IS. Matches a combination of the P2 field (high byte) and the P0 field (low byte); external address.

address. A number, reference, or expression. Although this entry is not literally an "address" when used with OPCODE, VALUE, P0, P1, or P2, the same kinds of entries are permitted as for addresses. See Address under Expression for details.

mask. A binary, octal, or hexadecimal number with one or more digits specified as "don't'care" (X). Don't-care digits match any values occurring in the processor data. Refer to the *Operating Instructions*, chapter 3, for details on masks.

NEWEST Command

Format:

```
NEWEST
```

Elements:

NEWEST. Moves display pointer in the trace buffer to just past the item most recently collected.

Example:

```
*NEWEST
*N                    ;Note abbreviation to N
```

MOVE Command

Format:

MOVE [[-] *number-of-lines*]

Elements:

MOVE. Moves the display pointer in the trace buffer. The display pointer controls the beginning or ending of the trace display. Movement is forward toward the most recent information, or (when "-" is used), backward toward the earliest information.

number-of-lines. A decimal number, or an expression in which the default for numeric constants is decimal. In Instructions mode, one display line is one instruction; in Frames mode, one display line is one frame.

For more details, see NEWEST Command, OLDEST Command, PRINT Command, TRACE Mode Command.

Examples:

```
*MOVE                    ;Same as "MOVE 1".
*MOVE 10
*MOVE -25
```

OPCODE Command

Format:

```
OPCODE
```

Elements:

OPCODE. Displays the hexadecimal opcode of the last instruction in the trace buffer. (The value is 00H if the buffer is empty.)

Example:

```
*OPCODE
*OPC                    ;Note abbreviation.
```

NOTE

The OPCODE is updated after each break in emulation if a LOCATION frame can be found within the last 16 frames of trace information. See Error Messages, warning message CA, for more details.

OLDEST Command

Format:

```
OLDEST
```

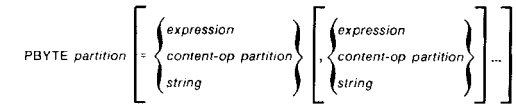
Elements:

OLDEST. Moves the display pointer in the trace buffer to just before the earliest entry.

Example:

```
*OLDEST
*O                    ;Note abbreviation to O.
```

Format:



Elements:

PBYTE. Displays or sets the contents of one or more address in external data memory. The range for PBYTE is 0 to 64K.

partition. A single address, or a range of addresses expresses as *address* TO *address* or *address* LENGTH *address*. See Partition.

If the data requires more memory than the size of the partition, an error occurs. If the data requires less memory, the data is repeated until the partition is filled.

expression. A number, reference, or a formula that evaluates to a number. See Expression.

content-op. One of the content operators CBYTE (code memory), DBYTE (data memory), RBYTE (register memory), XBYTE (external data memory, verified), PBYTE (unverified external data memory), or RBIT (bit-addressable memory). See Expression.

string. One or more characters enclosed in apostrophes ('). The content is the ASCII value of each character in successive bytes.

Examples:

```
*PBYTE 0100H              ;Display contents

*PBYTE C000 TO            ;Set partition to zero
  CFFFH = 0

*PBY F000H = DBY          ;Copy a table
  .TABLE TO .TABLE
  .+ 50H

*PBY E000H = 'ABCDEF',    ;Using a string in a list of
  80H, DBY 30H            ;values.
  LENGTH 20H
```

NOTE

Changes to memory using PBYTE are not verified by the emulator system. If read-after-write verification is desired, use XBYTE instead of PBYTE (see XBYTE commands.)

Partition

Format:

$$\left\{\begin{array}{l} address \\ address1 \text{ TO } address2 \\ address1 \text{ LENGTH } address2 \end{array}\right\}$$

Elements:

address. A number, reference, or expression. (See Address under Expression.)

TO. Describes a range of addresses. With TO, the entry *address1* is the lowest address in the range, and *address2* is the highest address in the range.

LENGTH. Describes a range of addresses. With LENGTH, *address1* is the lowest address in the range, and *address2* is the number of addresses in the range.

Examples (Using CBYTE command, format CBYTE *partition*):

```
*CBYTE 0                ;One address.
*CBYTE 0 TO 0FH         ;Sixteen consecutive addresses.
*CBYTE 0 LENGTH 10H     ;Also sixteen consecutive
                        ;addresses.
```

Format:

```
PPC
```

Elements:

PPC. Displays the address of the opcode of the last instruction collected in the trace buffer.

Examples:

```
*PPC
```

NOTE

The values of PPC is updated after each break in emulation, if a LOCATION frame can be found in the last 16 frames of trace information. See Error Messages, warning CA for details.

PC Command

Format:

PC [= *expression*]

Elements:

PC. Displays or sets the 16-bit program counter.

expression. A number, reference, or formula that evaluates to a number (see Expression).

Examples:

```
*PC                 ;Display program counter.
*PC = .START        ;Set PC to new address
```

Format:

```
PUT  :Fn:filename { MACRO                                          }
                  { :macro-name [ , :macro-name ] ...              }
```

Elements

PUT. Stores one or more macro definitions into an ISIS-II file.

:F*n*:*filename*. The drive (*n* = 0 to 9) and name of the file that is to receive the macro definitions. If the file does not already exist on that drive, the system opens the file for output. If the file does already exist, the previous contents are overwritten by the PUT command.

MACRO. Stores all macro definitions currently in the macro table.

:*macro-name*. The name of a macro preceded by a colon, or a list of such names separated by commas. The definitions of the macros in the list are output to file.

Examples:

```
*PUT :F1:JUL21.MAC MACRO
*PUT :F2:JUL21A.MAC :GOER
*PUT :F1:JUL21B.MAC :M, :RPT
```

PRINT Command

Format:

```
PRINT [ ALL                  ]
      [ [-] number of lines  ]
```

Elements:

PRINT. Displays one or more items from the trace buffer.

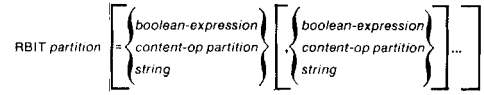
ALL. Displays all instructions or all frames (depending on the trace display mode), from the oldest to the newest.

number-of-lines. A decimal number, or an expression in which the default for numeric constants is decimal. In Instructions mode, each display line contains information for one instruction; in Frames mode, each display line contains one frame. When number-of-lines is positive, display starts with the line at the display pointer. When number-of-lines is negative (by using "-"), display ends with the line right before the pointer.

Examples:

```
*PRINT          ;Equivalent to "PRINT 1".
*PRINT ALL      ;Displays entime buffer.
*P 10           ;Note abbreviation.
*P -5
```

Format:



Elements:

RBIT. Displays or sets the contents of one or more addresses in bit-addressable memory.

partition. A single address, or a range of addresses expressed as *address* TO *address* or *address* LENGTH *address*; see Partition. For bit addresses, the range is 00H to FFH inclusive, but not all bit-addresses in that range exist in memory. If the data requires more memory than the size of the partition, an error occurs. If the data requires less memory, the data is repeated until the partition is filled.

boolean-expression. A number, reference, or expression. The result is truncated to its least significant bit (see Expression).

content-op. One of the content operators CBYTE (code memory), DBYTE (on-chip data memory), RBYTE (register memory), XBYTE (external data memory, verified) PBYTE (external data memory, unverified), or RBIT (bit-addressable memory).

string. One or more characters enclosed in apostrophes ('). The value for RBIT is the least significant bit of the ASCII value of each character, in successive bits.

Examples:

```
*RBIT .CY                    ;Display bit setting.
*RBIT 0 TO 7FH = 0
*RBIT .CY = NOT RBIT .PSW + 5
```

QR, QR0, QR1 Commands

Format:

$$\left\{\begin{matrix}\text{QR0}\\\text{QR1}\\\text{QR}\end{matrix}\right\} \; [\; = \textit{unlimited-match-condition} \;]$$

Elements:

QR0. Displays or sets trace qualifier register QR0.

QR1. Displays or sets trace qualifier register QR1.

QR. Displays both QR0 and QR1, or sets both QR0 and QR1 to the same match condition.

unlimited-match-condition. Specifies one or more non-overlapping fields of processor data to use for matching, and gives a single value, a masked value, or a partition (range) of values to use for each field. (If a partition is used, the system may adjust the partition; see *Unlimited Match Condition* for details.) When a qualifier register condition matches a frame of processor data, and the qualifier register is enabled, the next frame is collected in the trace buffer. Using the trace modes, qualifier register matches can turn trace on for 1000 frames or turn trace off (see *TR Commands*).

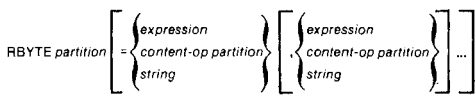
Examples:

```
*QR0                          ;Displays QR0 match condition
*QR1 = 0100H TO 01FFH
*QR1 = VALUE IS 00H TO
   1FH AND P1 IS 20H
*QR = XADDR IS C000H
   TO CFFFH
```

NOTES

Initially, both qualifier registers are set to all don't-cares (match every frame). The QR, QR0, and QR1 commands do not reset any values previously set in the specified register; only the lists specified in the (unlimited) match condition and affected (see the RESET commands for details on resetting breakpoint registers).

Format:



Elements.

RBYTE. Displays or sets the contents of one or more addresses.

partition. A single address, or a range of addresses expresses as *address* TO *address* or *address* LENGTH *address*. See Partition. For RBYTE, addresses are 80H to FFH, inclusive; some addresses in this range do not exist in memory. If the data requires more memory than the size of the partition, an error occurs. If the data requires less memory, the data is repeated until the partition is filled.

expression. A number, reference, or a formula that evaluates to a number. See Expression.

content-op. One of the content operators CBYTE (code memory), DBYTE (data memory), RBYTE (register memory), XBYTE (external memory, verified), PBYTE (external data memory, unverified), or RBIT (bit-addressable memory). See Expression.

string. One or more characters enclosed in apostrophes ('). The content is the ASCII value of each character is successive bytes.

Examples:

```
*RBYTE .SP                ;Display contents
*RBYTE 80H to FFH = 0
*RBYTE .ACC = DBYTE .TABLE + 5
```

RBS Command

Format:

RBS [= *expression*]

Elements:

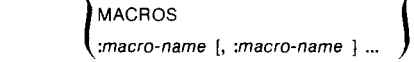
RBS. Displays or sets the register bank select (bits 3 and 4 of PSW). RBS governs the location of working registers R0 through R7 in data memory.

expression. A number, reference, or formula that evaluates to a number. For RBS, the result must be 0, 1, 2, or 3, or an error results.

Examples:

```
*RBS                ;Display current bank number.
*RBS = 3            ;Set RBS to bank 3.
```

Format:



Elements:

REMOVE. Deletes user-defined symbols or macros from the symbol table or macro definition table, respectively.

SYMBOLS. Removes all user-defined symbols.

.symbol-name. The name of a user defined symbol preceded by a period, or a list of such symbols separated by commas. The symbols in the list are removed. If two or more symbols exist with the same name, only the first one is removed.

MACROS. Removes all macro definitions. This command may not appear within any other command.

:macro-name. The name of a macro preceded by a colon (:), or a list of such macro names separated by commas. The macros in the list are removed. This command may not appear within any compound command.

Examples:

```
*REMOVE SYMBOLS      ;Remove all user symbols.
*REMOVE .START       ;Remove one symbol.
*REM .START, .LOOP   ;Note abbreviation to REM.
*REM MACROS          ;Remove all macro definitions.
*REM :GOER           ;Remove one macro.
;REM :M, :RPT        ;Remove list of macros.
```

REGISTERS Command

Format:

```
REGISTERS
```

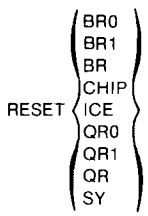
Elements:

REGISTERS. Displays contents of program counter (PC), accumulator (ACC), multiply register (B), stack pointer (SP), data pointer (DPTR), working registers R0 and R1 in the bank currently selected, and program status register (PSW). PSW is displayed in binary radix; the other registers are displayed in hexadecimal.

Example:

```
*REGISTERS          ;Display status register
*R                  ;Note abbreviation.
```

Format:



Elements:

RESET. Restores its object to the initial status that emulator invocation produces.

BR0. Resets breakpoint register BR0 to don't-care on all bits.

BR1. Resets breakpoint register BR1 to don't-care on all bits.

BR. Resets both BR0 and BR1 to don't-care on all bits.

CHIP. Resets the emulation processor. The stack pointer is reset to 7H; ports P0, P1, P2, and P3 and reset to 0FFH; all other on-chip registers are reset to 00H, including the Interrupt In Progress flag, which cannot be changed directly in any other way from the emulator.

ICE. Resets the emulator hardware; resets the map to 0,1000H. This command attempts recovery from an emulator hardware error state.

QR0. Resets trace qualifier register QR0 to don't-care on all bits.

QR1. Resets trace qualifier register QR1 to don't-care on all bits.

QR. Resets both QR0 and QR1 to don't-care on all bits.

SY. Disables SY0 OUT, SY0 LATCH, SY1 OUT, and SY1 LATCH.

Examples:

```
*RES CHIP          ;Note abbreviation RES.
*RES QR            ;Resets both QR0 and QR1, but
                   ;does not disable them.
*RES SY            ;See DISABLE commands for
                   ;details.
```

REPEAT Command

Format:

```
REPEAT cr
[ command cr                    ]
[ WHILE boolean-expression cr   ] ...
[ UNTIL boolean-expression cr   ]
ENDREPEAT
```

Elements:

REPEAT. Introduces a block of commands to be repeated.

cr. Intermediate carriage return; ends each input line but does not terminate the command entry. The carriage return after the END keyword is the termination of the command.

command. Any emulator command except DEFINE Macro and REMOVE Macro.

WHILE. Introduces a halt condition for the command block. The loop halts when the WHILE condition is tested and found to be FALSE.

UNTIL. Introduces a halt condition for the command block. The loop halts when the UNTIL condition is tested and found to be TRUE.

boolean-expression. Any number, reference, or expression (see Expression). The result is TRUE when the least significant bit of the result is 1, FALSE otherwise.

ENDREPEAT. Terminates entry of the command block. (May be abbreviated to END).

Example:

```
*REPEAT                 ;Introduce block to be repeated.
.*STEP                  ;Execute and display one
.*PRINT -1              ;instruction per repeat.
.*UNTIL PC = .DONE      ;Halt at end of program.
.*END                   ;End REPEAT.
```

NOTE

As the example shows, the system indicates you are within a block by displaying a period (.) before the prompt (one period for each block currently active).

Format:

SAVE :F*n*:*filename* [*partition* | NOCODE] [NOSYMBOLS]

Elements:

SAVE. Copies the user program (in absolute object file format) and user symbol table from code memory to ISIS-II file. No system symbols are saved.

:F*n*:*filename*. The drive (*n* = 0 - 9) and name of the file that is to receive the program. If the file does not already exist on the drive, the system opens the file for output. If the file does exist, it is overwritten by the SAVE Command.

partition. A single address, or a range of addresses expressed as *address* TO *address* or *address* LENGTH *address*; see Partition. If *partition* is not specified (and the NOCODE option is not used), the partition of code affected by the most recent LOAD or SAVE command is saved.

NOCODE. Suppresses saving the code (saves the symbol table only). NOCODE and *partition* are mutually exclusive in a given SAVE command.

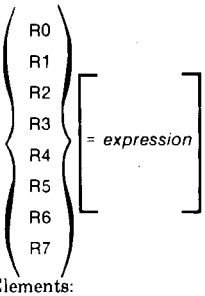
NOSYMBOLS. Suppresses saving the symbol table (saves code only).

Examples:

```
*SAVE :F1:PROG.OBJ
*SAVE :F2:PROG.OBJ 0 TO 3FH
*SAVE :F1:PROG.OBJ NOCODE
*SAVE :F2:PROG.OBJ NOSYMBOLS
```

R0 - R7 Commands

Format:



Elements:

R0 - R7. Display or set the contents of working registers in the currently selected register bank (see *RBS Command*).

expression. A number, reference, or formula that evaluates to a number. For R0 - R7, the result is truncated to the low byte.

Examples:

```
*R0                          ;Display register contents.
*R1 = 0                      ;Set register to 0.
*R7 = .COUNTER / 256T        ;Using a formula.
```

Format:

STACK [*expression*]

Elements:

STACK. Displays the contents of the emulation processor stack area and stack pointer value.

expression. Indicates the number of bytes at the top of the stack to be displayed. If the result exceeds the value of (SP + 1), the display terminates at data location 0. If the expression is omitted, the display starts at the current stack top and stops at data location 0.

Examples:

```
*STACK 5
*STA                    ;Note abbreviation
```

SECONDS Command

Format:

```
SECONDS
```

Elements:

SECONDS. Displays the 32-bit half-microsecond emulation timer (combination of TIMER and HTIMER) translated into microseconds.

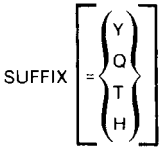
Example:

```
*SECONDS
*SEC                    ;Note abbreviation
```

NOTE

The emulation timer is cleared whenever the program counter changes. It is disabled and cleared during single step emulation. The timer is accurate to approximately ±0.01%; however, breaking emulation produces a "roundoff" error of ±1 μsec.

Format:



Elements:

SUFFIX. Displays or sets the default radix for numbers entered at the console. (See the *BASE Command.*)

Y. Binary radix.

Q. Octal radix (letter "O" may also be used for octal).

T. Decimal radix.

H. Hexadecimal radix. Hexadecimal is the initial default.

Examples:

```
*SUFFIX          ;Display current radix.
*SUF = Y         ;Set to binary radix.
*SUF = H         ;Note abbreviation to SUF.
```

STEP Command

Format:

STEP [FROM *address*]

Elements:

STEP. Executes one instruction, then halts.

FROM. Sets the program counter to the address of the instruction to be executed. If the FROM clause is omitted, the current program counter is used.

address. A number, reference, or expression. (see Address under Expression).

Examples:

```
*STEP
*STEP FROM .START
*S F 0100H          ;Note abbreviations
                    ;S (for STEP) and
                    ;F (for FROM).
```

NOTE

During single step emulation trace is always collected; external signal SY0 IN is ignored; the emulation timer is cleared and disabled.

SYMBOLS Command

Format:

```
SYMBOLS
```

Elements:

SYMBOLS. Displays all the user-defined symbols and their corresponding values.

Examples:

```
*SYMBOLS          ;Display symbol table
*SYM              :Note abbreviation.
```

SY, SY0, SY1 Display Commands

Format:

$$\left\{\begin{matrix}\text{SY}\\\text{SY0}\\\text{SY1}\end{matrix}\right\}$$

Elements:

SY. Displays the level of SY0 IN, the enabled/disabled status of SY0 OUT and SY0 LATCH, the level of SY1 IN, and the enabled/disabled status of SY1 OUT and SY1 LATCH.

SY0. Displays the level of SY0 as a bit value (0 or 1).

SY1. Displays the level of SY1 as a bit value (0 or 1).

Examples:

```
*SY
*SY0
*SY1
```

The following three tables list the system-defined symbols for code addresses, register addresses, and bit addresses. These system symbols may be referenced in expressions and displayed individually, but they are not displayed by the SYMBOLS command; they may not be changed or removed. They are not loaded or saved with the user code.

System Symbols for Code Addresses

System Symbol	Address	Type of Interrupt
.RESET	00H	Power-On Reset
.EXTI0	03H	External Interrupt 0
.TIMER0	0BH	Timer 0 Interrupt
.EXTI1	13H	External Interrupt 1
.TIMER1	1BH	Timer 1 Interrupt
.SINT	23H	Serial Port Interrupt

System Symbols for Registers

System Symbol	Register Address (Hex)	Meaning
.P0	80H	Port 0
.SP	81H	Stack Pointer
.DPL	82H	Data Pointer, Low Byte
.DPH	83H	Data Pointer, High Byte
.TCON	88H	Timer Control
.PCON	87H	Baud Rate Control (ICE-44 only)
.TMOD	89H	Timer Mode
.TL0	8AH	Timer 0, Low Byte
.TL1	8BH	Timer 1, Low Byte
.TH0	8CH	Timer 0, High Byte
.TH1	8DH	Timer 1, High Byte
.P1	90H	Port 1
.SCON	98H	Serial Port Control (ICE-51 only)
.SBUF	99H	Serial Port Buffer (ICE-51 only)
.P2	A0H	Port 2
.IE	A8H	Interrupt Enable
.P3	B0H	Port 3
.IP	B8H	Interrupt Priority
.STS	C8H	Status Hardware Register (ICE-44 only)
.SMD	C9H	Serial Mode Hardware Register (ICE-44 only)

Symbolic Reference Commands

Format:

$$\left\{\begin{array}{l} .system\text{-}symbol \\ .user\text{-}symbol\ [\ .user\text{-}symbol\ ]\ ... \\ .user\text{-}symbol\ =\ expression \end{array}\right\}$$

Elements:

.system-symbol. The name of a system symbol preceded by a period. Displays the corresponding address in code memory, register memory, or bit-addressable memory. See System Symbols.

.user-symbol. The name of a user defined symbol preceded by a period. Displays the value from the user symbol table corresponding to the first instance of that symbol. User symbol names may be concatenated to form multiple display references (for example, the reference ".X.Y" means "the first instance of symbol .Y that follows the occurrence of .X in the symbol table.")

expression. A number, reference, or a formula that evaluates to a number. See Expression.

Examples:

```
*.RESET
*.START
*.START = 0106H
```

The following three tables list the system-defined symbols for code addresses, register addresses, and bit addresses. These system symbols may be referenced in expressions and displayed individually, but they are not displayed by the SYMBOLS command; they may not be changed or removed. They are not loaded or saved with the user code.

System Symbols for Code Addresses

System Symbol	Address	Type of Interrupt
.RESET	00H	Power-On Reset
.EXTI0	03H	External Interrupt 0
.TIMER0	0BH	Timer 0 Interrupt
.EXTI1	13H	External Interrupt 1
.TIMER1	1BH	Timer 1 Interrupt
.SINT	23H	Serial Port Interrupt

System Symbols for Registers

System Symbol	Register Address (Hex)	Meaning
.P0	80H	Port 0
.SP	81H	Stack Pointer
.DPL	82H	Data Pointer, Low Byte
.DPH	83H	Data Pointer, High Byte
.TCON	88H	Timer Control
.PCON	87H	Baud Rate Control (ICE-44 only)
.TMOD	89H	Timer Mode
.TL0	8AH	Timer 0, Low Byte
.TL1	8BH	Timer 1, Low Byte
.TH0	8CH	Timer 0, High Byte
.TH1	8DH	Timer 1, High Byte
.P1	90H	Port 1
.SCON	98H	Serial Port Control (ICE-51 only)
.SBUF	99H	Serial Port Buffer (ICE-51 only)
.P2	A0H	Port 2
.IE	A8H	Interrupt Enable
.P3	B0H	Port 3
.IP	B8H	Interrupt Priority
.STS	C8H	Status Hardware Register (ICE-44 only)
.SMD	C9H	Serial Mode Hardware Register (ICE-44 only)

System Symbols for Registers

System Symbol	Register Address (Hex)	Meaning
.RCB	CAH	Received Control Byte (ICE-44 only)
.RBL	CBH	Receive Buffer Length (ICE-44 only)
.RBS	CCH	Receive Buffer Start (ICE-44 only)
.RFL	CDH	Received Field Length (ICE-44 only)
.STAD	CEH	Station Address (ICE-44 only)
.DMACNT	CFH	DMA Count (ICE-44 only)
.PSW	D0H	Program Status Word
.NSNR	D8H	Send Count/ Receive Count Hardware Register (ICE-44 only)
.SIUST	D9H	SIU Controller State Counter (ICE-44 only)
.TCB	DAH	Transmit Control Byte (ICE-44 only)
.TBL	DBH	Transmit Buffer Length (ICE-44 only)
.TBS	DCH	Transmit Buffer Start (ICE-44 only)
.FIFO0	DDH	FIFO Byte Zero (ICE-44 only)
.FIFO1	DEH	FIFO Byte One (ICE-44 only)
.FIFO2	DFH	FIFO Byte Two (ICE-44 only)
.ACC	E0H	Accumulator
.B	F0H	Multiplication Register

System Symbols for Bit Addresses

System Symbol	Hex Address	Meaning
.IT0	88H	Timer 0 Interrupt, Type Control Bit
.IE0	89H	Timer 0 Interrupt, Edge Flag
.IT1	8AH	Timer 1 Interrupt, Type Control Bit
.IE1	8BH	Timer 1 Interrupt, Edge Flag
.TR0	8CH	Timer 0 Run Control Bit
.TF0	8DH	Timer 0 Overflow Flag
.TR1	8EH	Timer 1 Run Control Bit
.TF1	8FH	Timer 1 Overflow Flag
.RI	98H	Receive Interrupt Flag (ICE-51 only)
.TI	99H	Transmit Interrupt Flag (ICE-51 only)
.RB8	9AH	Receive Bit 8 (ICE-51 only)
.TB8	9BH	Transmit Bit 8 (ICE-51 only)

.REN	9CH	Receiver Enable (ICE-51 only)
.SM2	9DH	Serial Mode Control Bit 2 (ICE-51 only)
.SM1	9EH	Serial Mode Control Bit 1 (ICE-51 only)
.SM0	9FH	Serial Mode Control Bit 2 (ICE-51 only)
.EX0	A8H	Enable External Interrupt 0
.ET0	A9H	Enable Timer 0 Interrupt
.EX1	AAH	Enable External Interrupt 1
.ET1	ABH	Enable Timer 1 Interrupt
.ES	ACH	Enable Serial Port Interrupt
.EA	AFH	Enable All Interrupts
.RXD	B0H	Serial Port Receive Pin
.TXD	B1H	Serial Port Transmit Pin
.INT0	B2H	Interrupt 0 Input Pin
.INT1	B3H	Interrupt 1 Input Pin
.T0	B4H	Timer/Counter 0 External Flag
.T1	B5H	Timer/Counter 1 External Flag
.WR	B6H	Write Data (for External Memory)
.RD	B7H	Read Data (for External Memory)
.PX0	B8H	Priority of External Interrupt 0
.PT0	B9H	Priority of Timer 0 Interrupt
.PX1	BAH	Priority of External Interrupt 1
.PT1	BBH	Priority of Timer 1 Interrupt
.PS	BCH	Priority of Serial Interrupt
.RBP	C8H	Receive Buffer Protect (ICE-44 only)
.AM	C9H	Auto Mode/ Addressed Mode (ICE-44 only)
.OPB	CAH	Optional Poll Bit (ICE-44 only)
.BOV	CBH	Receive Information Buffer Overflow (ICE-44 only)
.SI	CCH	SIU Interrupt (ICE-44 only)
.RTS	CDH	Request To Send (ICE-44 only)
.RBE	CEH	Receive Buffer Empty (ICE-44 only)
.TBF	CFH	Transmit Buffer Full (ICE-44 only)
.P	D0H	Parity Flag
.OV	D2H	Overflow Flag
.RS0	D3H	Register Bank Select Bit 0
.RS1	D4H	Register Bank Select Bit 1
.F0	D5H	Flag 0
.AC	D6H	Auxiliary Carry Flag
.CY	D7H	Carry Flag
.SER	D8H	Sequence Error Receive (ICE-44 only)
.NR0	D9H	Receive Sequence Counter 0 (ICE-44 only)

System Symbols for Bit Addresses

System Symbol	Hex Address	Meaning
.NR1	DAH	Receive Sequence Counter 1 (ICE-44 only)
.NR2	DBH	Receive Sequence Counter 2 (ICE-44 only)
.SES	DCH	Sequence Error Send (ICE-44 only)
.NS0	DDH	Send Sequence Counter 0 (ICE-44 only)
.NSI	DEH	Send Sequence Counter 1 (ICE-44 only)
.NS2	DFH	Send Sequence Counter 2 (ICE-44 only)

TIMER Command

Format:

```
TIMER
```

Elements:

TIMER. Displays the value of the low 16 bits of the 32-bit half-microsecond emulation timer. The emulation timer is cleared whenever the program counter changes, and during single-step emulation. (See also HTIMER Command, SECONDS Command).

Example:

```
*TIMER
```

TM0, TM1 Commands

Format:

```
{TM0}
{TM1} [ = expression ]
```

Elements:

TM0. Displays or sets the value of the 16-bit on-chip timer TIMER 0.

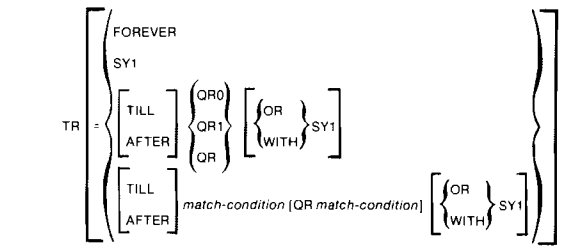
TM1. Displays or sets the value of the 16-bit on-chip timer TIMER 1.

expression. A number, reference, or formula that evaluates to a number (see Expression).

Examples:

```
*TM0                ;Display timer value.
*TM1 = 0            ;Clear timer
*TM0 = .INIT        ;Load initial value.
```

TR Commands



Elements:

TR. Displays or sets the trace register controls for the collection of trace information.

FOREVER. Sets the trace register to collect every frame (no controls enabled).

SY1. Enables external signal SY1 IN to control trace. When SY1 is enabled (and any other enabled factors are satisfied), trace is collected while SY1 is high, and stops when SY1 goes low (if SY1 LATCH is enabled, trace does not restart if SY1 goes high again; otherwise, SY1 can toggle trace on and off. See ENABLE Commands and DISABLE Commands).

TILL. A trigger mode that enables a match by an enabled qualifier register (QR0 or QR1, as specified in the same TR command as the TILL) to halt trace collection. The buffer retains the last 1000 frames.

AFTER. A trigger mode that enables a match by the combination of qualifier registers specified in the TR command to start trace, collect 1000 frames, then halt automatically, unless emulation breaks earlier.

QR0. Enables trace qualifier register QR0 to control trace (depending on the trigger mode, if any) by matching the processor data frame by frame.

QR1. Enables trace qualifier register QR1 to control trace (depending on the trigger mode, if any) by matching the processor data frame by frame.

QR. Enables both QR0 and QR1.

WITH SY1. Enables a match by a combination of qualifier registers and SY1 IN going high simultaneously to control trace (the exact effect depends on the trigger mode, if any).

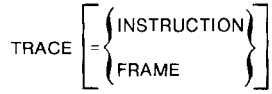
OR SY1. Enables qualifier register match or SY1 IN going high to control trace (the exact effect depends on the trigger mode if any). (For details, refer to the Operating Instructions, chapter 6.)

match-condition. Specifies one or more non-overlapping fields of processor data to use for matching, and gives a single value or a mask value to use for each designated field (see Match Condition). If one match condition is given, it is set in QR0 and QR0 is enabled; if *match-condition* OR *match-condition* is given, the first match condition is set in QR0 and the second condition is set in QR1, and both QR0 and QR1 are enabled.

Examples:

```
*TR                          ;Display current setting of TR
                             ;and QR's.
*TR = FOREVER
*TR = SY1
*TR = QR0
*TR = AFTER QR1
*TR = TILL QR OR SY1
*TR = AFTER .START OR 07FFH WITH SY1
```

Format:



Elements:

TRACE. Displays or sets the trace display mode. The trace display mode determines whether one line of trace display shall be one instruction or one frame, and whether the MOVE command shall move the pointer by instructions or by frames. (See PRINT Command, MOVE Command.)

INSTRUCTION. Sets one display line and one MOVE line to be one instruction. INSTRUCTION is the initial mode.

FRAME. Sets one display line and one MOVE line to be one frame.

Examples:

```
*TRACE                    ;Display current mode.
*TRACE = INSTRUCTION      ;Initial display mode.
*TRA = FRA                ;Note abbreviation.
```

Trace Reference

Format:

FRAME *trace-group*

Elements:

FRAME. As an operand in an expression, references the field of trace information corresponding to the trace group, in the frame pointed to by the trace buffer display pointer. (An error occurs if the buffer is empty or if the pointer is at NEWEST.) (see *Expression.*)

trace-group. One of the following references to fields of trace information:

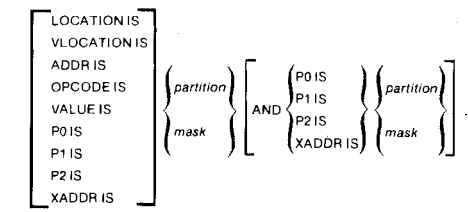
ADDR	16 bits	— Address if address frame.
DATA	8 bits	— Low order byte of ADDR, data if data frame
DMUX	1 bit	— 0 if address frame, 1 if data frame
CYC	1 bit	— 0 if first fetch, 1 if second, third or fourth fetch
P0	8 bits	— Port 0
P1	8 bits	— Port 1
P2	8 bits	— Port 2
XADDR	16 bits	— Port 2 concatenated with Port 0
TOVF	1 bit	— Trace overflow flag

Examples (using the EVALUATE command, format EVALUATE *trace-reference*):

```
*EVALUATE FRAME ADDR      ;Displays value of address field
*EVA FRA P0               ;Note abbreviation to FRA
```

Unlimited Match Condition

Format:



Elements:

LOCATION IS. Specifies that the address or mask value is to be used to match the code address field on LOCATION frames only (address of opcode). LOCATION IS is the default if no frame type is designated in the match condition.

VLOCATION IS. Matches the code address field on VLOCATION (operand address) frames only.

ADDR IS. Matches the code address field on both LOCATION and VLOCATION frames.

OPCODE IS. Matches the data field (low byte of address field) on OPCODE frames only.

VALUE IS. Matches the data field (low byte of address field) on VALUE frames (operand value) only.

P0 IS. Matches the P0 field.

P1 IS. Matches the P1 field.

P2 IS. Matches the P2 field.

XADDR IS. Matches a combination of the P2 field (high byte) and the P0 field (low byte); external address.

partition. A single address (or other value) or a range of addresses (values) expressed as *address* TO *address* or *address* LENGTH *address*. The system adjusts the low and high bounds of a match partition to preserve the highest-order difference; all lower nibbles are adjusted to 0H (low-bound) and FH (high-bound), respectively.

mask. A binary, octal, or hexadecimal number with one or more digits specified as "don't care" (X). Don't care digits match any values occurring in the processor data. Refer to the *Operating Instructions*, chapter 3, for details on masks.

AND. Combines a match on P0, P1, P2, or XADDR with the condition specified in the previous clause (or clauses) of the match condition. More than one port can be ANDed to form a combination. To produce a match, the values specified must all occur on the same frame.

Examples (using the BR0 command, format BR0 = *unlimited-match-condition*):

```
*BR0 = LOCATION IS 0 TO 10H

*BR0 = VALUE IS F7H TO FFH

*BR0 = P0 IS 0 TO 0FH and P1 IS 10H to 1FH
```

NOTES

When matching on instructions that change ports P0, P1, or P2, note that the changed value of the port does not appear until the next OPCODE frame following the execution of the instruction that changed the port; this limitation applies only to the individual ports, not to ports used as an external address (XADDR).

The Unlimited Match Condition allows partitions of address and other values to be specified for matching; this form is allowed when setting the breakpoint or trace qualifier registers directly. By contrast, the forms of match condition allowed with the GO, GR, and TR commands do not permit partitions (see Match Condition). In effect, the system must have a chance to adjust the partition before the register can be enabled.

User Symbol Commands

The following commands operate on the user symbol table:

DEFINE Symbol Command.

REMOVE Symbol Command.

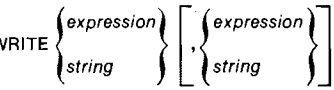
Symbolic Reference Commands

SYMBOLS Command

Also see: LOAD Command. SAVE Command.

WRITE Command

Format:



Elements:

WRITE. Displays an (evaluated) expression or a string of characters (or a sequence of such elements) on the system console.

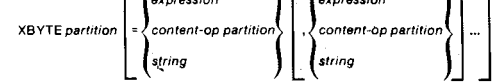
expression. A number, reference, or a formula that evaluates to a number (see Expression). For WRITE, the expression is evaluated and the result is displayed in the current BASE.

string. A sequence of characters enclosed in apostrophes ('). To include the apostrophe character itself in the string, use a double apostrophe (' ').

Examples:

```
*WRITE 'HELLO, THERE!'          ;Display message
*WRITE 'THE VALUE IS:', .VALUE
                                 ;Combine text and expression
*WRI 'cr'                        ;Outputs carriage return
                                 ;(blank line).
```

Format:



Elements:

XBYTE. Displays or sets the contents of one or more addresses in external data memory; the range for XBYTE is 0 to 64K.

partition. A single address, or a range of addresses expresses as *address* TO *address* or *address* LENGTH *address*. See Partition. If the data requires more memory than the size of the partition, an error occurs. If the data requires less memory, the data is repeated until the partition is filled.

expression. A number, reference, or a formula that evaluates to a number. See *Expression.*

content-op. One of the content operators CBYTE (code memory), DBYTE (data memory), RBYTE (register memory), XBYTE (external data memory, verified), PBYTE (unverified external data memory), or RBIT (bit-addressable memory). See *Expression.*

string. One or more characters enclosed in apostrophes ('). The content is the ASCII value of each character in successive bytes.

Examples:

```
*XBYTE 0100H                ;Display contents

*XBYTE C000 TO              ;Set partitions to zero.
  CFFFH = 0

*XBY F000H = DBY            ;Copy a table.
  .TABLE TO .TABLE
  + 50H

*XBY E000H = 'ABCDEF',      ;Using a string and other
  80H, DBY .30H             ;values in a list.
    LENGTH 20H
```

NOTE

Changes to memory using XBYTE are verified by read-after-write. If verification is not desired, use PBYTE instead of XBYTE (see PBYTE Commands).

3065 Bowers Avenue, Santa Clara, California 95051
(408) 987-8080

Printed in U.S.A.

DS-015/2.5K/0483/AP